Interreg
Italy - Croatia
S.LI.DES.
European Regional Development Fund

EUROPEAN UNION

S.LI.DES.
Smart strategies for sustainable tourism in LIvely cultural DEStinations

2014 - 2020 Interreg V-A
Italy - Croatia CBC Programme
Priority Axis: Environment and cultural heritage
Specific objective: 3.1 - Make natural and cultural heritage a leverage for sustainable and more balanced territorial development

Deliverable 3.1.3.

Datahub User Manual

Work Package:	**3 – The S.LI.DES. Smart Destination Ecosystem**
Activity:	**1 – Building the Destination data hub**
Responsible Partner:	**PP7 – Institute for Tourism**
Partners involved:	**LP – Ca' Foscari University of Venice (IT)** **PP1 – CISET (IT)** **PP2 – Ecipa (IT)** **PP3 – SIPRO Ferrara (IT)** **PP4 – City of Bari (IT)** **PP5 – City of Venice (IT)** **PP6 – CAST – University of Bologna (IT)** **PP7 – Institute for Tourism** **PP8 – Craft College – Institution for adult education Subsidiary Rijeka** **PP9 – Development Agency of the City of Dubrovnik – Dura** **PP10 – Šibenik Tourist board**

Version:	**Final V.01**	**Date**	**06/28/2021**
Type:	**Report**		
Availability:	**Public**		
Editors:	**LP – Ca' Foscari University of Venice, PP2 – Ecipa, PP7 – Institute for Tourism**		

Table of Contents

INTRODUCTION

The S.LI.DES. project aims at fostering cross-border cooperation among cultural destinations in the Programme area of Interreg ITA-HR and the joint planning of smart strategies to support more sustainable and balanced territorial development through the promotion of tangible and intangible cultural heritage.

The operational tool chosen to reach this goal is an interactive dashboard, through which five destinations in the Adriatic (and its representative stakeholders) could manage their tourist product information, the tourist offer, and the tourist anthropic pressure in a unique personalized place that collect input from multiple channels and sources.

The datahub will help cities stakeholders to:

- be aware of and monitor their tourism and non-tourism performances from different urban perspectives (economic, social, environmental, cultural, mobility, etc.) (activity 3.1);
- be aware of and monitor the mobility of visitors within the city, especially of pedestrian mobility, and relate it to the location of POIs (in particular, of tangible and intangible heritage) and to the organization of events (activity 3.2);
- within tangible and intangible heritage, mapping craft activities and Culture&Creative Industries (CCIs) and defining a sort of "ranking" of these activities according to the tourism potential (activity 3.3).

The analysis is based on different data sources such as data provided from the destination, from social media, and from other public and private bodies that deal with tourism.

The challenges were multiple: from collecting as much data as possible to the creation of tourism indicators, from the involvement of artisans, and to the conversion in a user-friendly visualization.

The user should consider the datahub as a "box" where all data were collected and harmonize and linked to each other based on tourism indicators.

The data collection phase, such as sources and indicators is deeply explained and analyzed in the deliverables D3.1.1_Datahub Framework and D3.1.2_Datahub_prototype.

This manual represents a technical guide addresses to the developer.

1. The Operational Workflow

This short introduction shows the design and construction of the data hub.

The process was divided into three parts:
1 - Partners sent data to DB to the development team. Data come from different sources in a different formats. The researchers' team worked to find the right data and tried to harmonize them with each other.
2 - Data collected by each partner.
In this second stage, data was preprocessing to fit right to DB design (Data processing phase).
3 - Some data come from in-real time sources: the development team worked to build a web service to link different sources into one data hub.

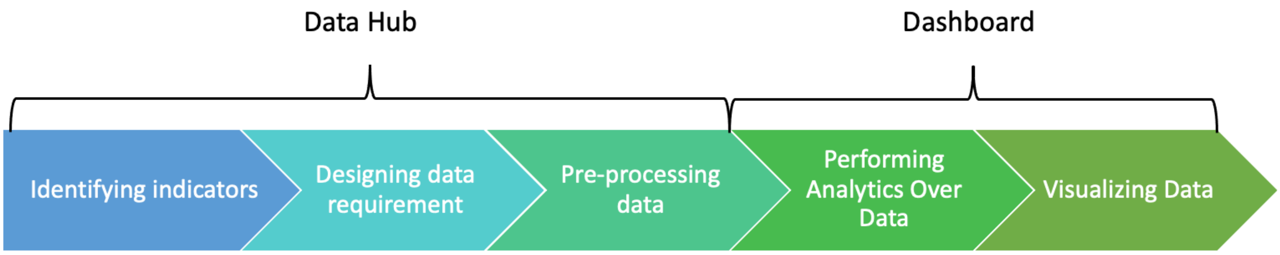


Figure 1. Operational Workflow

2. Data sources

This section indicates the main data sources currently in use.

2.1. Non-relational DB

The datahub's main source is a non-relational DB. This choice (non-relational) is based on 2 main factors:

1) expected size of data (millions of rows)
2) discrepancy of data structures

The sources feeding the DB are:

1) remote sources directly writing valid data,
2) remote sources calling web services that validate data and write on DB,
3) una tantum import of static sources (which cannot be made automatic and must be manually revised).

2.2. Relational DB

A second DB (this time relational) was chosen, to respond to 3 needs:

1) get more efficient access to a reworked and reordered subset of the non-relational DB's data,
2) manage some sources whose structures were better suited to be managed with this kind of DB,
3) Manage ACL.

The sources feeding the DB are:

1) re-elaborations of non-relational DB's data,
2) remote sources directly writing valid data.

2.3. Remote Sources

To represent the mobility data detected by the real-time sensors, an external remote web service is also used. Please find more details in document D3.2.1_Mobility_Database and document D3.2.2_Now-casting_model.

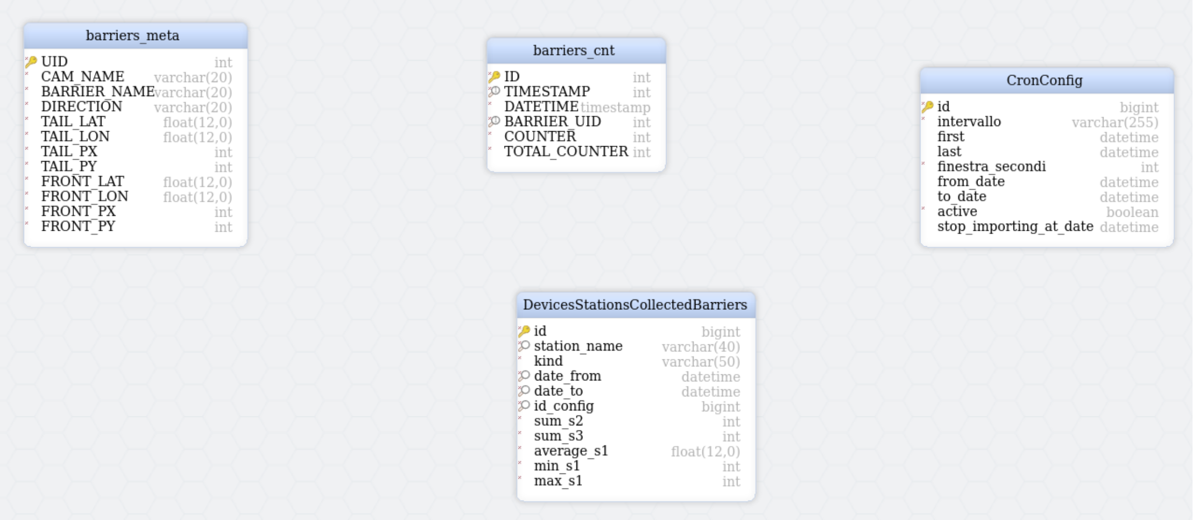
2.4. Json Files

It is possible to provide the dashboard "dynamic" source via a coordinated format json files.

Please see next chapter 3.3 to deepen this topic.

2.5. SQL schema

2.5.1. Bari



barriers_meta	
UID	int
CAM_NAME	varchar(20)
BARRIER_NAME	varchar(20)
DIRECTION	varchar(20)
TAIL_LAT	float(12,0)
TAIL_LON	float(12,0)
TAIL_PX	int
TAIL_PY	int
FRONT_LAT	float(12,0)
FRONT_LON	float(12,0)
FRONT_PX	int
FRONT_PY	int

barriers_cnt	
ID	int
TIMESTAMP	int
DATETIME	timestamp
BARRIER_UID	int
COUNTER	int
TOTAL_COUNTER	int

CronConfig	
id	bigint
intervallo	varchar(255)
first	datetime
last	datetime
finestra_secondi	int
from_date	datetime
to_date	datetime
active	boolean
stop_importing_at_date	datetime

DevicesStationsCollectedBarriers	
id	bigint
station_name	varchar(40)
kind	varchar(50)
date_from	datetime
date_to	datetime
id_config	bigint
sum_s2	int
sum_s3	int
average_s1	float(12,0)
min_s1	int
max_s1	int

Figure 2. SQL schema for the city of Bari

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	intervallo	varchar(255)	linux crontab notation to define when the script runs. minutes hours day month year. * * * * * means every minute of every day */5 * * * * means every 5 minutes of every day and so on
	first	datetime	
	last	datetime	

*	finestra_secondi	int	a number of seconds for aggregation. */5 * * * * with 300 means that every 5 minutes the system will aggregate 300 seconds of events
	from_date	datetime	start to run cron from this date
	to_date	datetime	stop to run cron after this date
*	active	boolean DEFAULT false	
	stop_importing_at_date	datetime	do not import events occurring after this date
Indexes			
	pk_cronconfig	ON id	

Table 1. CronConfig (Handles cronjob's configurations)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	unique id
*	station_name	varchar(40)	name of the station (cam)
*	kind	varchar(50)	ackword compatibility
*	date_from	datetime	define the start of aggregation window (aggregation windows id defined by CronConfi:finestra_secondi)

*	date_to	datetime	*define the end of aggregation window (aggregation windows id defined by CronConfi:finestra_secondi)*
*	id_config	bigint	refers to CronConfig table
*	sum_s2	int	sum of all the counters for a single station having meta barrier = S2 during the “from/to” interval
*	sum_s3	int	sum of all the counters for a single station having meta barrier = S3 during the “from/to” interval
*	average_s1	float(12,0)	average value of all the counter for a single station having meta barrier = s1 durgin the “from / to” interval
*	min_s1	int	owest value of all the counters for a single station having meta barrier = s1 during the “from/to” interval
*	max_s1	int	the highest value of all the counters for a single station having meta barrier = s1 during the “from/to” interval
Indexes			
	pk_devicesstationscollectedbarriers	ON id	
	date_from	ON date_from	
	date_to	ON date_to	

	id_config	ON id_config	
	station_name	ON station_name	

Table 2. Table DevicesStationsCollectedBarriers ("Barriers_cnt" aggregation following CronConfig rules: s1 points the number of people detected; s2 and s3 points people's flow)

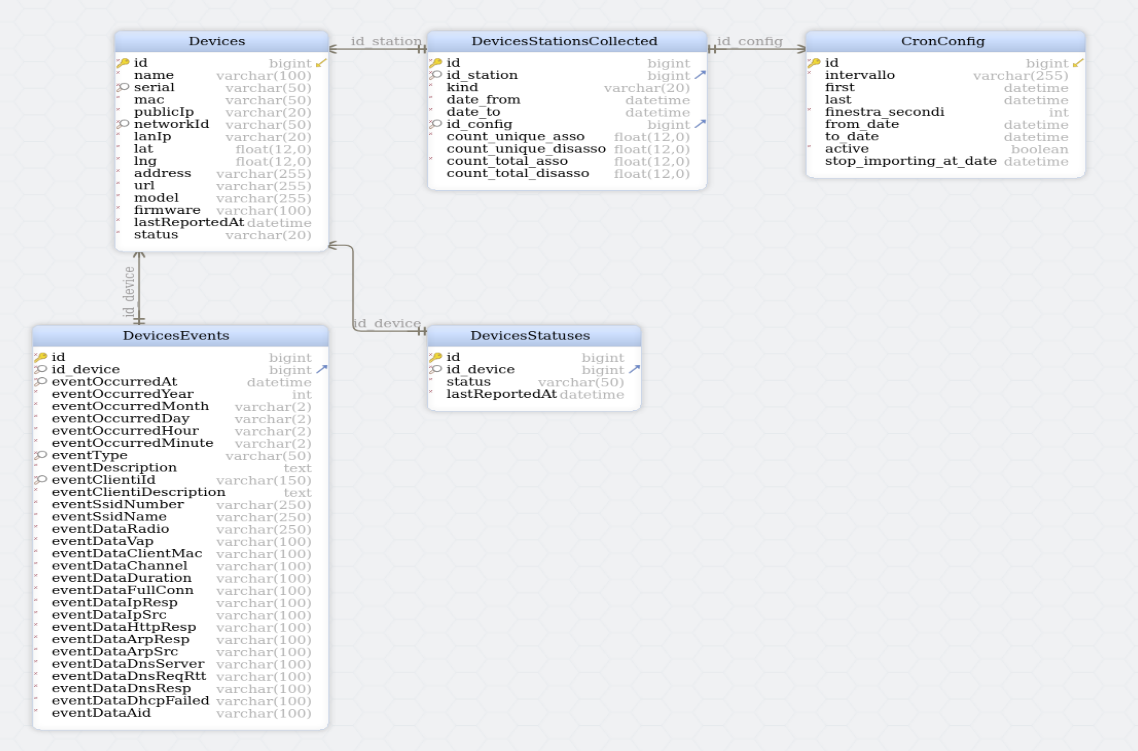
Idx	**Field Name**	**Data Type**	**Description**
*	ID	int AUTO_INCREMENT	record internal id
*	TIMESTAMP	int UNSIGNED	record unix timestamp
*	DATETIME	timestamp DEFAULT CURRENT_TIMESTAMP	record datetime UTC (use TIMESTAMP for queries)
*	BARRIER_UID	int	barrier unique id (join with barriers_meta for queries)
*	COUNTER	int	*number of people detected during the event*
*	TOTAL_COUNTER	int	*total of barrier's people ever detected*
Indexes			
	pk_barriers_cnt	ON ID	
	UID	ON TIMESTAMP, BARRIER_UID	
	BARRIER_UID	ON BARRIER_UID	

Table 3. Barriers_cnt (Collects data about each barrier in relation with barriers_meta (to define if it is counting or direction, which barrier and which cam)

Idx	Field Name	Data Type	Description
*	UID	int AUTO_INCREMENT	unique id
*	CAM_NAME	varchar(20)	*name of the single cam*
*	BARRIER_NAME	varchar(20)	*name of the barrier (cam network)*
*	DIRECTION	varchar(20)	S1 (ount) S2 (irection 1) S3 (Direction 2)
*	TAIL_LAT	float(12,0)	barrier TAIL point Latitude (EPSG:4326)
*	TAIL_LON	float(12,0)	barrier TAIL point Longitude (EPSG:4326)
*	TAIL_PX	int	barrier TAIL point pixel X coordinate
*	TAIL_PY	int	barrier TAIL point pixel Y coordinate
*	FRONT_LAT	float(12,0)	barrier FRONT point Latitude (EPSG:4326)
*	FRONT_LON	float(12,0)	barrier FRONT point Longitude (EPSG:4326)
*	FRONT_PX	int	barrier FRONT point pixel X coordinate
*	FRONT_PY	int	barrier FRONT point pixel Y coordinate
		Indexes	
	pk_barriers_meta	ON UID	

Table 4. Barriers meta (Meta information about each barrier: kind (counting or direction), cam name and position)

2.5.2. Dubrovnik



Devices	
id	bigint
name	varchar(100)
serial	varchar(50)
mac	varchar(50)
publicIp	varchar(20)
networkId	varchar(50)
lanIp	varchar(20)
lat	float(12,0)
lng	float(12,0)
address	varchar(255)
url	varchar(255)
model	varchar(255)
firmware	varchar(100)
lastReportedAt	datetime
status	varchar(20)

DevicesStationsCollected	
id	bigint
id_station	bigint
kind	varchar(20)
date_from	datetime
date_to	datetime
id_config	bigint
count_unique_asso	float(12,0)
count_unique_disasso	float(12,0)
count_total_asso	float(12,0)
count_total_disasso	float(12,0)

CronConfig	
id	bigint
intervallo	varchar(255)
first	datetime
last	datetime
finestra_secondi	int
from_date	datetime
to_date	datetime
active	boolean
stop_importing_at_date	datetime

DevicesEvents	
id	bigint
id_device	bigint
eventOccurredAt	datetime
eventOccurredYear	int
eventOccurredMonth	varchar(2)
eventOccurredDay	varchar(2)
eventOccurredHour	varchar(2)
eventOccurredMinute	varchar(2)
eventType	varchar(50)
eventDescription	text
eventClientiId	varchar(150)
eventClientiDescription	text
eventSsidNumber	varchar(250)
eventSsidName	varchar(250)
eventDataRadio	varchar(250)
eventDataVap	varchar(100)
eventDataClientMac	varchar(100)
eventDataChannel	varchar(100)
eventDataDuration	varchar(100)
eventDataFullConn	varchar(100)
eventDataIpResp	varchar(100)
eventDataIpSrc	varchar(100)
eventDataHttpResp	varchar(100)
eventDataArpResp	varchar(100)
eventDataArpSrc	varchar(100)
eventDataDnsServer	varchar(100)
eventDataDnsReqRtt	varchar(100)
eventDataDnsResp	varchar(100)
eventDataDhcpFailed	varchar(100)
eventDataAid	varchar(100)

DevicesStatuses	
id	bigint
id_device	bigint
status	varchar(50)
lastReportedAt	datetime

Figure 3. SQL schema for the city of Dubrovnik

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	intervallo	varchar(255)	linux crontab notation to define when the script runs. minutes hours day month year. * * * * * means every minute of every day */5 * * * * means every 5 minutes of every day and so on

	first	datetime	
	last	datetime	
*	finestra_secondi	int	number of seconds for aggregation. */5 * * * * with 300 means that every 5 minutes the system will aggregate 300 seconds of events
	from_date	datetime	start to run cron from this date
	to_date	datetime	stop to run cron after this date
*	active	boolean DEFAULT false	
	stop_importing_at_date	datetime	do not import events occurring after this date
		Indexes	
	pk_cronconfig	ON id	

Table 5. CronConfig (Handles cronjob's configurations)

Idx	Field Name	Data Type
*	id	bigint AUTO_INCREMENT
*	name	varchar(100)
*	serial	varchar(50)
*	mac	varchar(50)
*	publicIp	varchar(20)
*	networkId	varchar(50)

*	lanIp	varchar(20)
*	lat	float(12,0)
*	lng	float(12,0)
*	address	varchar(255)
*	url	varchar(255)
*	model	varchar(255)
*	firmware	varchar(100)
*	lastReportedAt	datetime
*	status	varchar(20)
Indexes		
	pk_devices	ON id
	networkId	ON networkId
	serial	ON serial

Table 6. Devices (Meta information about every active collecting device (name, position, and network)

Idx	Field Name	Data Type
*	id	bigint AUTO_INCREMENT
*	id_device	bigint
*	eventOccurredAt	datetime
*	eventOccurredYear	int
*	eventOccurredMonth	varchar(2)
*	eventOccurredDay	varchar(2)
*	eventOccurredHour	varchar(2)
*	eventOccurredMinute	varchar(2)

*	eventType	varchar(50)
*	eventDescription	text
*	eventClientiId	varchar(150)
*	eventClientiDescription	text
*	eventSsidNumber	varchar(250)
*	eventSsidName	varchar(250)
*	eventDataRadio	varchar(250)
*	eventDataVap	varchar(100)
*	eventDataClientMac	varchar(100)
*	eventDataChannel	varchar(100)
*	eventDataDuration	varchar(100)
*	eventDataFullConn	varchar(100)
*	eventDataIpResp	varchar(100)
*	eventDataIpSrc	varchar(100)
*	eventDataHttpResp	varchar(100)
*	eventDataArpResp	varchar(100)
*	eventDataArpSrc	varchar(100)
*	eventDataDnsServer	varchar(100)
*	eventDataDnsReqRtt	varchar(100)
*	eventDataDnsResp	varchar(100)
*	eventDataDhcpFailed	varchar(100)
*	eventDataAid	varchar(100)
Indexes		
	pk_devicesevents	ON id
	eventClientiId	ON eventClientiId

	eventOccurredAt	ON eventOccurredAt
	eventType	ON eventType
	id_device	ON id_device
Foreign Keys		
	DevicesEvents_ibfk_1	(id_device) ref Devices (id)

Table 7. DevicesEvents (Collects data from each event related to devices in Dubrovnik's project)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	id_station	bigint	
*	kind	varchar(20)	*for backward compatibility with other tables. Always: association*
*	date_from	datetime	*define the start of aggregation window (aggregation windows id defined by CronConfi:finestra_secondi)*
*	date_to	datetime	*define the end of the aggregation window (aggregation windows id defined by CronConfi:finestra_secondi)*
*	id_config	bigint	
*	count_unique_asso	float(12,0)	*sum the number of single devices association*
	count_unique_disasso	float(12,0)	*sum the number of single devices association*

*	count_total_asso	float(12,0)	*sum the number of devices association (the same device can be counted several times)*
	count_total_disasso	float(12,0)	*sum the number of devices disassociation (the same device can be counted several times)*
Indexes			
	pk_devicesstationscollected	ON id	
	id_config	ON id_config	
	id_station	ON id_station	
Foreign Keys			
	DevicesStationsCollected_ibfk_2	(id_config) ref CronConfig (id)	
	DevicesStationsCollected_ibfk_1	(id_station) ref Devices (id)	

Table 8. DevicesStationsCollected (Collects data from devices. Name's fields explain the use.)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	id_device	bigint	*refers to Devices table*
*	status	varchar(50)	*defines the devices's status (online or offline)*
*	lastReportedAt	datetime	
Indexes			
	pk_devicesstatuses	ON id	

	DevicesStatuses_ibfk_1	ON id_device	
Foreign Keys			
	DevicesStatuses_ibfk_1	(id_device) ref Devices (id)	

Table 9. DevicesStatuses (Handles list of devices' status)

2.5.3. Ferrara

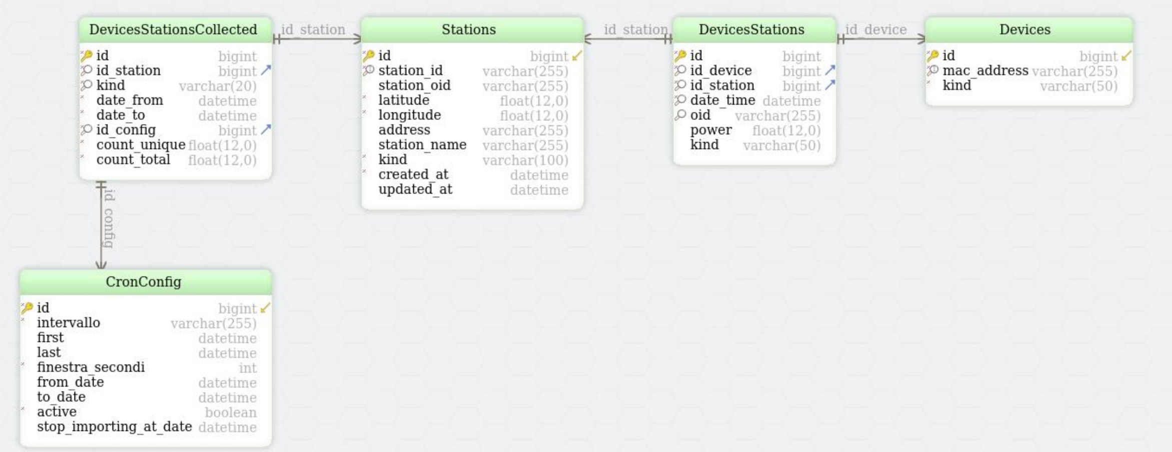


Figure 4. SQL schema for the city of Ferrara

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	intervallo	varchar(255)	linux crontab notation to define when the script runs. minutes hours day month year. * * * * * means every minute of every day */5 * * * * means every 5

			minutes of every day and so on
	first	datetime	
	last	datetime	
*	finestra_secondi	int	a number of seconds for aggregation. */5 * * * * with 300 means that every 5 minutes the system will aggregate 300 seconds of events
	from_date	datetime	start to run cron from this date
	to_date	datetime	stop to run cron after this date
*	active	boolean DEFAULT false	
	stop_importing_at_date	datetime	do not import events occurring after this date
		Indexes	
	pk_cronconfig	ON id	

Table 10. CronConfig (Handles cronjob's configurations)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	mac_address	varchar(255)	*mac address of the device*
*	kind	varchar(50)	*wifi or bluetooth*
		Indexes	

	pk_devices	ON id	
	mac_address	ON mac_address	

Table 11. Devices (List of every device that has ever hooked up a station (Wi-Fi or Bluetooth)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	id_device	bigint	refers to Devices table
*	id_station	bigint	refers to Stations table
*	date_time	datetime	when event occurred
	oid	varchar(255)	single event identifier
	power	float(12,0)	
	kind	varchar(50)	bluetooth or wifi
	Indexes		
	pk_devicesstations	ON id	
	date_time	ON date_time	
	id_device	ON id_device	
	id_station	ON id_station	
	oid	ON oid	
	Foreign Keys		
	DevicesStations_ibfk_1	(id_device) ref Devices (id)	
	DevicesStations_ibfk_2	(id_station) ref Stations (id)	

Table 12. DevicesStations (Connection table of each device on each station)

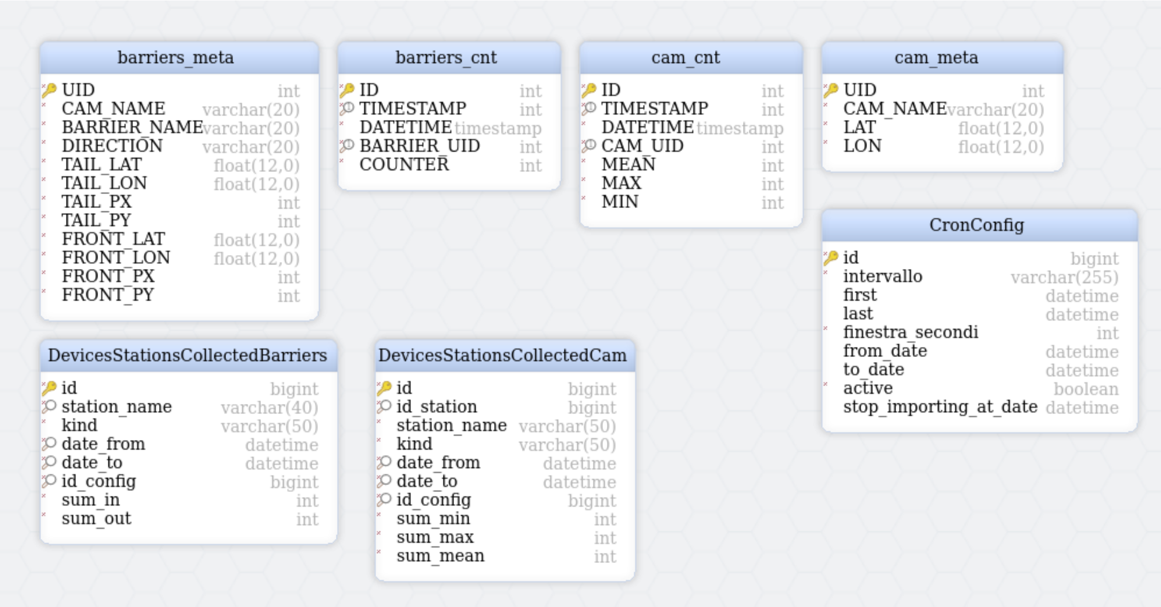
Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	id_station	bigint	*refers to Stations table*
*	kind	varchar(20)	*identifies between bluetooth (bt) or wifi (wifi)*
*	date_from	datetime	
*	date_to	datetime	
*	id_config	bigint	*refers to CronConfig table*
*	count_unique	float(12,0)	*sum the number of single devices association*
*	count_total	float(12,0)	*sum the number of devices association (the same device can be counted several times)*
		Indexes	
	pk_devicesstationscollected	ON id	
	id_config	ON id_config	
	id_station	ON id_station	
	kind	ON kind	
		Foreign Keys	
	id_config	(id_config) ref CronConfig (id)	
	id_station	(id_station) ref Stations (id)	

Table 13. DevicesStationsCollected (DeviceEvents aggregation following CronConfig rules)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	station_id	varchar(255)	*station internal identifier*
	station_oid	varchar(255)	*alphanumeric station identifier*
*	latitude	float(12,0)	
*	longitude	float(12,0)	
	address	varchar(255)	
	station_name	varchar(255)	
*	kind	varchar(100)	*bluetooth or wifi*
*	created_at	datetime DEFAULT CURRENT_TIMESTAMP	
	updated_at	datetime	
		Indexes	
	pk_stations	ON id	
	station_id	ON station_id	

Table 14. Stations (List of each station in the city)

2.5.4. Šibenik



barriers_meta	
UID	int
CAM_NAME	varchar(20)
BARRIER_NAME	varchar(20)
DIRECTION	varchar(20)
TAIL_LAT	float(12,0)
TAIL_LON	float(12,0)
TAIL_PX	int
TAIL_PY	int
FRONT_LAT	float(12,0)
FRONT_LON	float(12,0)
FRONT_PX	int
FRONT_PY	int

barriers_cnt	
ID	int
TIMESTAMP	int
DATETIME	timestamp
BARRIER_UID	int
COUNTER	int

cam_cnt	
ID	int
TIMESTAMP	int
DATETIME	timestamp
CAM_UID	int
MEAN	int
MAX	int
MIN	int

cam_meta	
UID	int
CAM_NAME	varchar(20)
LAT	float(12,0)
LON	float(12,0)

CronConfig	
id	bigint
intervallo	varchar(255)
first	datetime
last	datetime
finestra_secondi	int
from_date	datetime
to_date	datetime
active	boolean
stop_importing_at_date	datetime

DevicesStationsCollectedBarriers	
id	bigint
station_name	varchar(40)
kind	varchar(50)
date_from	datetime
date_to	datetime
id_config	bigint
sum_in	int
sum_out	int

DevicesStationsCollectedCam	
id	bigint
id_station	bigint
station_name	varchar(50)
kind	varchar(50)
date_from	datetime
date_to	datetime
id_config	bigint
sum_min	int
sum_max	int
sum_mean	int

Figure 5. SQL schema for the city of Šibenik

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	intervallo	varchar(255)	linux crontab notation to define when the script runs. minutes hours day month year. * * * * * means every minute of every day */5 * * * * means every 5 minutes of every day and so on
	first	datetime	
	last	datetime	
*	finestra_secondi	int	a number of seconds for aggregation.

			*/5 * * * * with 300 means that every 5 minutes the system will aggregate 300 seconds of events
	from_date	datetime	start to run cron from this date
	to_date	datetime	stop to run cron after this date
*	active	boolean DEFAULT false	
	stop_importing_at_date	datetime	do not import events occurring after this date
		Indexes	
	pk_cronconfig	ON id	

Table 15. CronConfig (Handles cronjob's configurations)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	station_name	varchar(40)	
*	kind	varchar(50)	*bluetooth or wifi*
*	date_from	datetime	
*	date_to	datetime	
*	id_config	bigint	
*	sum_in	int	*a sum of the detected event for a single barrier referring to the barrier with direction "in" for the defined window*

*	sum_out	int	*a sum of the detected event for a single barrier referring to the barrier with direction “out” for the defined window*
Indexes			
	pk_devicesstationscollectedbarriers	ON id	
	date_from	ON date_from	
	date_to	ON date_to	
	id_config	ON id_config	
	station_name	ON station_name	

Table 16. DevicesStationsCollectedBarriers (Barriers cnt aggregation following CronConfig rules)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	id_station	bigint	
*	station_name	varchar(50)	
*	kind	varchar(50)	
*	date_from	datetime	
*	date_to	datetime	
*	id_config	bigint	
*	sum_min	int	*the lowest number detected in the time windows*
*	sum_max	int	*the highest number detected in the time windows*

*	sum_mean	int	*the mean number detected in the time windows*
		Indexes	
	pk_devicesstationscollectedcam	ON id	
	date_from	ON date_from	
	date_to	ON date_to	
	id_config	ON id_config	
	id_station	ON id_station	

Table 17. DevicesStationsCollectedCam (Collects from Cam related to each stations' devices)

Idx	Field Name	Data Type	Description
*	ID	int AUTO_INCREMENT	Record internal id
*	TIMESTAMP	int UNSIGNED	Record unix timestamp
*	DATETIME	timestamp DEFAULT CURRENT_TIMESTAMP	Record datetime UTC (use TIMESTAMP for queries)
*	BARRIER_UID	int	Barrier unique id (join with barriers_meta for queries)
*	COUNTER	int	Counter value
		Indexes	
	pk_barriers_cnt	ON ID	
	UID	ON TIMESTAMP, BARRIER_UID	
	BARRIER_UID	ON BARRIER_UID	

Table 18. Barriers_cnt (Collects data about each barrier in relation with "barriers_meta" table (to define if it is counting or direction, which barrier and which cam)

Idx	Field Name	Data Type	Description
*	UID	int	Unique id
*	CAM_NAME	varchar(20)	Camera name
*	BARRIER_NAME	varchar(20)	Barrier Name
*	DIRECTION	varchar(20)	Direction IN/OUT
*	TAIL_LAT	float(12,0)	Barrier TAIL point Latitude (EPSG:4326)
*	TAIL_LON	float(12,0)	Barrier TAIL point Longitude (EPSG:4326)
*	TAIL_PX	int	Barrier TAIL point pixel X coordinate
*	TAIL_PY	int	Barrier TAIL point pixel Y coordinate
*	FRONT_LAT	float(12,0)	Barrier FRONT point Latitude (EPSG:4326)
*	FRONT_LON	float(12,0)	Barrier FRONT point Longitude (EPSG:4326)
*	FRONT_PX	int	Barrier FRONT point pixel X coordinate
*	FRONT_PY	int	Barrier FRONT point pixel Y coordinate
		Indexes	
	pk_barriers_meta	ON UID	

Table 19. Barriers_meta (Meta information about each barrier: kind counting or direction), cam name and position)

Idx	Field Name	Data Type	Description
*	ID	int AUTO_INCREMENT	Record internal id
*	TIMESTAMP	int UNSIGNED	Record unix timestamp
*	DATETIME	timestamp DEFAULT CURRENT_TIMESTAMP	Record datetime UTC (use TIMESTAMP for queries)

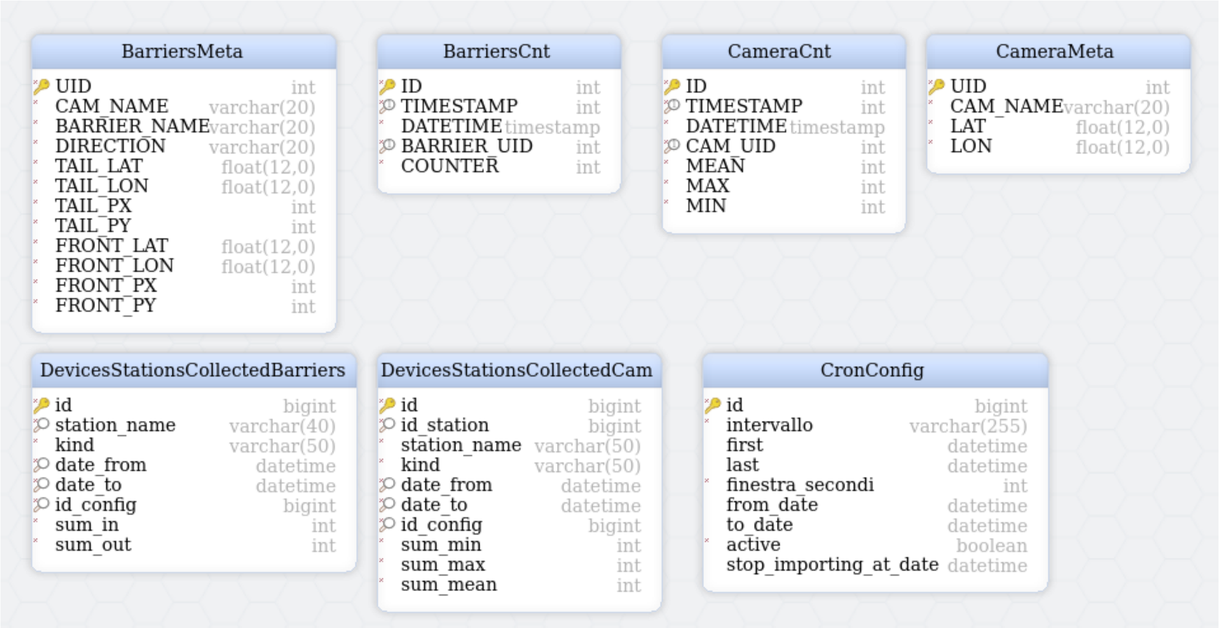
*	CAM_UID	int	Camera unique id (join with cam_meta for queries)
*	MEAN	int	Mean counter value
*	MAX	int	Max counter value
*	MIN	int	Min counter value
Indexes			
	pk_cam_cnt	ON ID	
	UID	ON TIMESTAMP, CAM_UID	
	CAM_UID	ON CAM_UID	

Table 20. Cam_cnt (Collects data about each cam in relation to barriers)

Idx	Field Name	Data Type	Description
*	UID	int	Unique id
*	CAM_NAME	varchar(20)	Camera Name
*	LAT	float(12,0)	Camera Latitude (EPSG:4326)
*	LON	float(12,0)	Camera Longitude (EPSG:4326)
Indexes			
	pk_cam_meta	ON UID	

Table 21. Cam_meta (List of all camera’s positions and name)

2.5.5. Venice



BarriersMeta	
UID	int
CAM_NAME	varchar(20)
BARRIER_NAME	varchar(20)
DIRECTION	varchar(20)
TAIL_LAT	float(12,0)
TAIL_LON	float(12,0)
TAIL_PX	int
TAIL_PY	int
FRONT_LAT	float(12,0)
FRONT_LON	float(12,0)
FRONT_PX	int
FRONT_PY	int

BarriersCnt	
ID	int
TIMESTAMP	int
DATETIME	timestamp
BARRIER_UID	int
COUNTER	int

CameraCnt	
ID	int
TIMESTAMP	int
DATETIME	timestamp
CAM_UID	int
MEAN	int
MAX	int
MIN	int

CameraMeta	
UID	int
CAM_NAME	varchar(20)
LAT	float(12,0)
LON	float(12,0)

DevicesStationsCollectedBarriers	
id	bigint
station_name	varchar(40)
kind	varchar(50)
date_from	datetime
date_to	datetime
id_config	bigint
sum_in	int
sum_out	int

DevicesStationsCollectedCam	
id	bigint
id_station	bigint
station_name	varchar(50)
kind	varchar(50)
date_from	datetime
date_to	datetime
id_config	bigint
sum_min	int
sum_max	int
sum_mean	int

CronConfig	
id	bigint
intervallo	varchar(255)
first	datetime
last	datetime
finestra_secondi	int
from_date	datetime
to_date	datetime
active	boolean
stop_importing_at_date	datetime

Figure 6. SQL schema for the city of Venezia

Idx	Field Name	Data Type	Description
*	ID	int AUTO_INCREMENT	Record internal id
*	TIMESTAMP	int UNSIGNED	Record unix timestamp
*	DATETIME	timestamp DEFAULT CURRENT_TIMESTAMP	Record datetime UTC (use TIMESTAMP for queries)
*	BARRIER_UID	int	Barrier unique id (join with barriers_meta for queries)
*	COUNTER	int	Counter value
		Indexes	
	pk_barrierscnt	ON ID	
	UID	ON TIMESTAMP, BARRIER_UID	

Table 22. BarriersCnt (Collects data about each barrier in relation with barriers_meta (to define if it is counting or direction, which barrier and which cam)

Idx	Field Name	Data Type	Description
*	UID	int	Unique id
*	CAM_NAME	varchar(20)	Camera name
*	BARRIER_NAME	varchar(20)	Barrier Name
*	DIRECTION	varchar(20)	Direction IN/OUT
*	TAIL_LAT	float(12,0)	Barrier TAIL point Latitude (EPSG:4326)
*	TAIL_LON	float(12,0)	Barrier TAIL point Longitude (EPSG:4326)
*	TAIL_PX	int	Barrier TAIL point pixel X coordinate
*	TAIL_PY	int	Barrier TAIL point pixel Y coordinate
*	FRONT_LAT	float(12,0)	Barrier FRONT point Latitude (EPSG:4326)
*	FRONT_LON	float(12,0)	Barrier FRONT point Longitude (EPSG:4326)

*	FRONT_PX	int	Barrier FRONT point pixel X coordinate
*	FRONT_PY	int	Barrier FRONT point pixel Y coordinate
		Indexes	
	pk_barriersmeta	ON UID	

Table 23. BarriersMeta (Meta information about each barrier: kind (counting or direction), cam name and position)

Idx	**Field Name**	**Data Type**	**Description**
*	ID	int AUTO_INCREMENT	Record internal id
*	TIMESTAMP	int UNSIGNED	Record unix timestamp
*	DATETIME	timestamp DEFAULT CURRENT_TIMESTAMP	Record datetime UTC (use TIMESTAMP for queries)
*	CAM_UID	int	Camera unique id (join with cam_meta for queries)
*	MEAN	int	Mean counter value
*	MAX	int	Max counter value
*	MIN	int	Min counter value
		Indexes	
	pk_cameracnt	ON ID	
	UID	ON TIMESTAMP, CAM_UID	

Table 24. CameraCnt (Collects data about each cam in relation to barriers)

Idx	Field Name	Data Type	Description
*	UID	int	Unique id
*	CAM_NAME	varchar(20)	Camera Name
*	LAT	float(12,0)	Camera Latitude (EPSG:4326)
*	LON	float(12,0)	Camera Longitude (EPSG:4326)
Indexes			
	pk_camerameta	ON UID	

Table 25. CameraMeta (List of all position’s camera)

Idx	Field Name	Data Type	Description
*	id	bigint AUTO_INCREMENT	
*	intervallo	varchar(255)	linux crontab notation to define when the script runs. minutes hours day month year. * * * * * means every minute of every day */5 * * * * means every 5 minutes of every day and so on
	first	datetime	
	last	datetime	

*	finestra_secondi	int	a number of seconds for aggregation. */5 * * * * with 300 means that every 5 minutes the system will aggregate 300 seconds of events
	from_date	datetime	start to run cron from this date
	to_date	datetime	stop to run cron after this date
*	active	boolean DEFAULT false	
	stop_importing_at_date	datetime	do not import events occurring after this date
Indexes			
	pk_cronconfig	ON id	

Table 26. CronConfig (Handles cronjob's configurations)

Idx	Field Name	Data Type
*	id	bigint AUTO_INCREMENT
*	station_name	varchar(40)
*	kind	varchar(50)
*	date_from	datetime
*	date_to	datetime
*	id_config	bigint
*	sum_in	int
*	sum_out	int
Indexes		

	pk_devicesstationscollectedbarriers	ON id
	date_from	ON date_from
	date_to	ON date_to
	id_config	ON id_config
	station_name	ON station_name

Table 27. DevicesStationsCollectedBarriers (Barriers_cnt aggregation following CronConfig rules)

Idx	Field Name	Data Type
*	id	bigint AUTO_INCREMENT
*	id_station	bigint
*	station_name	varchar(50)
*	kind	varchar(50)
*	date_from	datetime
*	date_to	datetime
*	id_config	bigint
*	sum_min	int
*	sum_max	int
*	sum_mean	int
	Indexes	
	pk_devicesstationscollectedcam	ON id
	date_from	ON date_from
	date_to	ON date_to
	id_config	ON id_config
	id_station	ON id_station

Table 28. DevicesStationsCollectedCam (DeviceEvents aggregation following CronConfig rules)

2.6. NoSQL

2.6.1. Bari PMA (People Movement Analyzer)*

**work in progress in this period since Bari just install the cameras*

List of events detected by cameras

```
{
        "_id"{"$oid":"609b9b80a00278467453e220"},
        "gmtoff":7200,
        "id_source":"local",
        "mac_address":"822fbbb058ba",
        "timestamp":"1620814205",
        "event_id":17,
        "sensor_name":"S2",
        "event_type":"Fire",
        "sensor_id":2,
        "message":"Fire",
        "description":"Fire",
        "frame_timestamp":5360000,
        "timezone":"CEST",
        "object"
                {
                        "top_left_x":0.046875,
                        "top_left_y":0.5314814814814814,
                        "width":0.019791666666666666,
                        "height":0.03888888888888889
                },
        "dash_id":"CJQCFpWO2Oamd"
}
```

Indexes:

_id

id_source

sensor_name

timestamp

id_source_sensor_name_timestamp

2.6.2. Dubrovnik PMA (People Movement Analyzer)

List of events detected by devices

```
{
        "_id":{"$oid":"60054c92835d510ca579de33"},
        "networkId":"N_659777345409800961",
        "deviceSerial":"Q2KD-Z2JC-Y7WM",
        "deviceName":"",
        "deviceMac":"e0:cb:bc:8d:69:d8",
        "devicePublicIp":"195.29.30.146",
        "deviceStatus":"online",
        "deviceLanIp":"192.168.200.4",
        "deviceLat":42.64037,
        "deviceLng":18.11075,
        "deviceAddress":"Pred Dvorom 3",
        "deviceModel":"MR42",
        "deviceFirmware":"wireless-25-11",
        "eventOccurredAt":"2021-01-18T08:52:52.930023Z",
        "eventOccurredYear":"2021",
        "eventOccurredMonth":"01",
        "eventOccurredDay":"18",
        "eventOccurredHour":"08",
        "eventOccurredMinute":"52",
        "eventType":"disassociation",
        "eventDescription":"802.11 disassociation",
        "eventClientiId":"k2f3ed4",
        "eventClientiDescription":"",
        "eventSsidNumber":0,
        "eventSsidName":"DUMUS",
        "eventDataRadio":"0",
        "eventDataVap":"0",
        "eventDataClientMac":"78:D7:5F:DD:F0:66",
        "eventDataChannel":"1",
        "eventDataDuration":"1562.308380621",
        "eventDataFullConn":"923.711340238",
        "eventDataIpResp":"923.711340238",
        "eventDataIpSrc":"10.47.62.212",
        "eventDataHttpResp":"928.659793839",
        "eventDataArpResp":"0.029990628",
        "eventDataArpSrc":"10.47.62.212",
        "eventDataDnsServer":"10.128.128.128",
        "eventDataDnsReqRtt":"0.009996876",
        "eventDataDnsResp":"1.819431427",
        "eventDataDhcpFailed":"1",
        "eventDataAid":"1947031354",
        "eventClientDescription":""
}
```

Indexes:

```
        _id
        date_time
        eventOccurredAt
        eventType
        networkId
        station_id
```

2.6.3. Ferrara PMA (People Movement Analyzer)

List of events detected by devices

```
{
  "_id":{"$oid":"5fc4f4b0cd87541cf8369ad6"},
   "mac-address":"dd5d346d73f95ee24fd50ee54d83f858ae81bcf873bef4ec216d4d5911e1e057",
  "power":-81,
  "station_id":"test_10",
  "timestamp":"1606739585.093794",
  "date_time":"2020-11-30 13:33:05.093794+01:00",
  "coordinates":[20.4,30.2],
  "random":true,
  "station_name":"test_10",
  "kind":"wifi",
  "kind_en":"wifi",
  "city":"Ferrara"
}
```

Indexes

_id
date_time
mac-address_station_id_date_time_kind_city
station_id

2.6.4. Šibenik and Venice

Sensor data by Sibenik and Venice are not stored in the NoSQL DB but in MariaDB. Please see section 2.5 SQL schema for further information.

Data

Please refer to section 4.3 “Queering a datahub” to view most of the available “schemas” minimum common.

```
 {
        "_id"{"$oid":"5e9b08daa23cac0aef1e0eb5"},
        "kind":"                                                                              ",
        "kind_en":"                                                                           ",
        "city":"cityName",
        "indicator":"sectionName",
        "file":"fileName",
        "values":[
                arrayOfValues
        ]
}
```

Feeds

Feeds from online newspapers and weather information.

```
{
        "_id":{"$oid":"5ed281513a534e047f18ea14"},
        "kind":"[weather|news]",
        "kind_en":"[weather|news]",
        "city":"[cityName]",
        "section":"feed",
        "day":"[YYYY-mm-dd]",
        "time":"[H:i]",
        "file":"[url]",
        "values":[
                valuesArray
        ]
}
```

Gtrends

Data collected from Gtrends service (imported manually from csv). Important: the data in Gtrend are not an absolute value, but it’s related to a specific index in a specific time range. So new import has to considerate all range time used before.

```
{
        "_id":{"$oid":"5dfa4e41505dca59c16dfda9"},
        "area":"All",
        "kind":"Termine di ricerca",
        "city":"[cityNa,e]",
        "from":[YYYY],
        "to":[YYYY],
        "categories":"[All | Viaggi]",
        "interval":"month",
        "year":[YYYY],
        "month":1..12,
        "value":0..100,
        "term":"[cityName]"
}
```

TripsAdvisor[CityName]

Each TripAdvisor table represents a reworking and aggregation of data collected automatically from the TripAdvisor service for some predefined points of interest.

The data refers to a time interval between 2020 and 2021.

```
{
        "_id":{"$oid":"606c8b739066ff34a63faca7"},
        "id":"[string]",
        "name":"[string]",
        "Y[2020|2021]":{
                "W[20..53]":{
                        "days":[1..7],
                        "average":{
                                "Reviews":0..N,
                                "Rating":0..5,
                                "Photo":0..N,
                                "Trips"{
```

```
                    "string":0..N,
                    …
                },
                "review_rating_count"{
                    "1":0..N,
                    "2":0..N,
                    "3":0..N,
                    "4":0..N,
                    "5":0..N
                },
                "rank":1..N,
                "rankof":1..N
            },
            "first_day":"YYYY-mm-dd",
            "last_day":"YYYY-mm-dd"
        }
        “groups":{
            "string":{
                "string":0..N,
                ...
            }, ...
        },
        "cats":{
            "string":["sting",...]
        }
    }
}
```

3. Adding data to a Datahub

This section describes how developers added data to the data hub and how to add new ones.

3.1. Static sources

Any information that a user loads into the system through a compiled file (often csv or xls) is a static source.

Since there is no standardization of data, for each non-automated data source it has been and will be necessary to create an "ad-hoc" import system.
e.g.: If you would like to enter the population data for the year 2021 (currently not present) it will therefore be necessary to agree on a template file format and provide the compiled file that will be imported later.

The system is potentially able to update or rewrite all sources deriving from static sources by simply uploading files with an agreed template.
Once the formats have been defined for each source and an interchange standard has been established, it should be possible for a user with permissions to add or modify the relevant datasets.

GoogleTrend
Google does not provide an automatic API system for downloading trend files.
This means that every imported trend has been manually imported and is, in fact, a static source.
The user has also to take into account in updating data that Google Trend always reviews on a range of values from 0 to 100 and rewrites the export on that interval.
If the user asks for 2004-2020 data, the max will be 100 and the min will be 0.
If the user asks for 2019-2020 data, the max will be 100 and the min will be 0.
This means that it is not possible to update the data by inserting just the "new year" but the entire reference period must necessarily be exported and imported rewriting all data.
The upload of updates for the year 2021 (to be done in 2022) will pass through the download of the google processing and subsequent upload to the system.

3.2. Dynamic sources

Any source capable of inserting information into the data hub without human intervention is considered a dynamic source.

Right now, the system is automatically recovering data from:

1) real-time sensors,
2) news RSS feeds,
3) weather RSS feeds.

On one hand, the definition of a completely new dynamic source (e.g. a system of API-calls for the exchange of data relating to arrivals and departures of people from the port) must be agreed upon and developed separately.

On the other hand, the list of feed sources (points 2 and 3 of the list above) can be easily updated through a simple request.

Regarding real-time sensors (point 1 of the list above), since there is no univocal standard (and it cannot be) among the various sources, an ad hoc data interchange system has been developed for each available source.

The system will automatically manage all new data sources within the same network of already existing sensors (it will simply receive new inputs in an already agreed format).

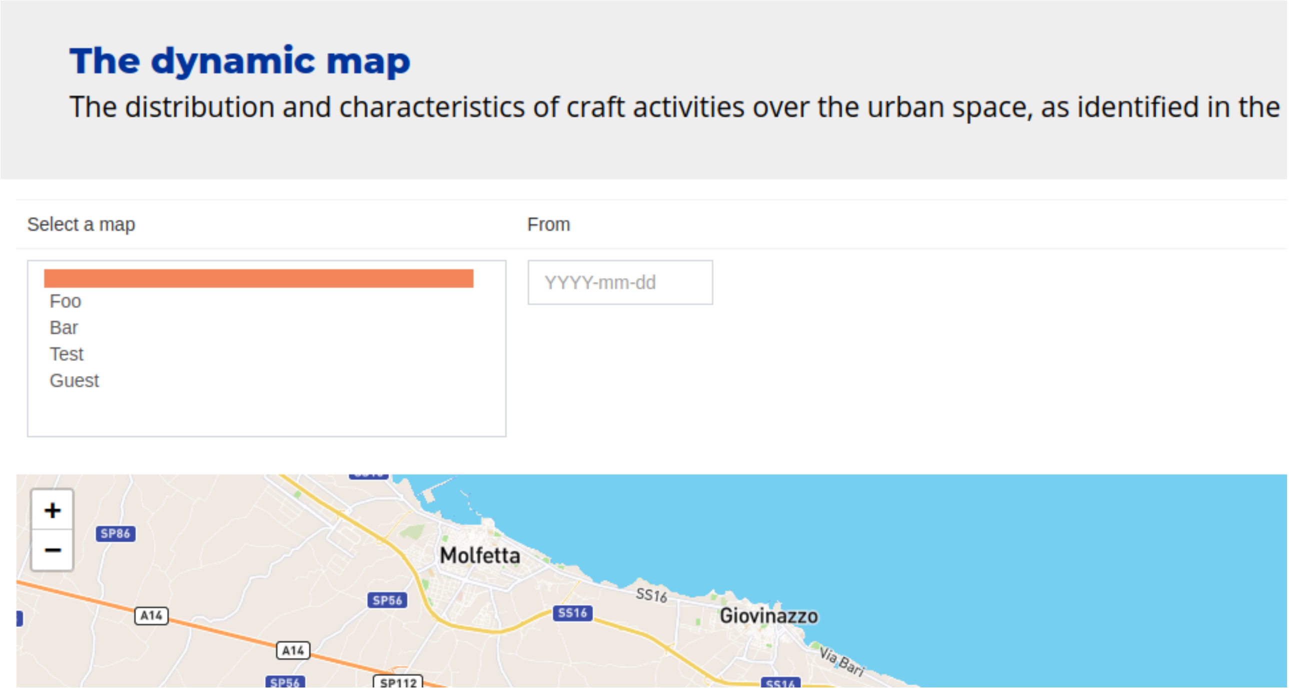
To insert a completely new detection network it is necessary to agree on a format, on an interchange protocol, and specific developments and analyzes.

3.3. File JSON

Defining two JSON files it is possible to dynamically insert data, define data sources, and see the result on the screen.

File1: map definition file

This file allows the user to define one or more maps that he/she can choose from the dashboard and on which to insert a series of markers and information.



The dynamic map

The distribution and characteristics of craft activities over the urban space, as identified in the

Select a map

From

Foo
Bar
Test
Guest

YYYY-mm-dd

Figure 7. Type of maps available at the Dashboard

FILE TYPOLOGY

The first section defines the type of file.

```
{
        ″table″: {
                ″name″: ″DynamicMapSets″,
                ″database″: ″DynamicMapSets″,
                        ″description″: ″Files that include the parameters of the possible dynamic maps″,
                        ″toMap″: true
 }
```

FIELD SECTION

The field section defines which and what kind of fields must or should be present in the map definitions.

```
″field″: [
{
        ″name″: ″Table″,
      ″description″: ″Table name″,
      ″visualizeOnMap″: false,
      ″visualizeOnTable″: false,
      ″valType″: ″string″,
      ″primaryKey″: false
},
{
      ″name″: ″City″,       //Bari, Dubrovnik …
      ″description″: ″City of interest″,
      ″visualizeOnMap″: true,
      ″visualizeOnTable″: true,
      ″valType″: ″shortText″,
      ″primaryKey″: false
},
{
      ″name″: ″MapName″,
      ″description″: ″Name of the map″, //Foo, Bar .. （ refer
                                   to the picture ）
      ″visualizeOnMap″: true,
      ″visualizeOnTable″: true,
      ″valType″: ″shortText″,
      "primaryKey": true
},
{
      "name": "POIs",
      "description": "Array of POIs to include in the map",
      "visualizeOnMap": true,
      "visualizeOnTable": true,
      "valType": "dataArray",
      "primaryKey": false
},
{
      "name": "MapID",
      "description": "List of the geographic coordinates of the NW-SE map corners",
      "visualizeOnMap": false,
      "visualizeOnTable": false,
      "valType": "dataArray",
      "primaryKey": false
},
],
```

RECORD SECTION

The record section defines the values for each map based on the previously defined fields.

```
"record": [
{
        "Table": "DynamicMapSets",
      "City": "ACity", // Bari, Dubrovnik ...
      ″MapName″: ″Foo″, // Foo, bar, test, guest
      ″POIs″: [
              {
                ″Id″: ″1″
              },
              {
                ″Id″: ″2″
              }
              ...
              {
                ″Id″:″N″
              }
      ],
      ″MapID″: [  // to understand this field simply refer to                          //
  “field”  section the field with
                        // name MapID and his description
              41.1323,
              16.8558,
              41.1183,
              16.8834
      ]
  },
  {
        ″Table″: ″DynamicMapSets″,
      ″City″: ″ACity″, // Bari, Dubrovnik ...
      ″MapName″: ″Bar″, // Foo, bar, test, guest
      ″POIs″: [
              {
                ″Id″: ″2″
              },
              {
                ″Id″: ″3″
              }
              ...
              {
                ″Id″:″M″
              }
       ],
       ″MapID″: [  // to understand this field simply refer to                        //
   “field”  section the field with
                         // name MapID and his description
              41.2323,
              16.9558,
              41.2183,
              16.7834
       ]
},

...
```

File 2: Information definition file

FILE TYPOLOGY

The first section defines the type of file.

```
{
    "table": {
          "name": "ExampleFile",
          "database": "Preliminary database + Stuff",
          "description": "POIs for the city of Foo",
          "toMap": true
    },
```

FIELD SECTION

In the field section, you define all the information that the user is going to transmit (pass).

Mandatory: identifier, latitude, and longitude.

The identifier is used to uniquely recognize a point and can be anything as long as it is unique in the file (usually an alphanumeric string).

The “user” field defines the minimum ACL required to view the data on the screen.

```
"field": [
      {
        "name": "Table",
        "description": "Service field that includes the Table name",
        "user": "A",
        "visualizeOnMap": false,
        "visualizeOnTable": false,
        "valType": "string",
        "primaryKey": false
      },
      {
        "name": "Date",
        "comment": "Date in which the questionary was submitted",
        "user": "A",
        ″visualizeOnMap″: false,
        ″visualizeOnTable″: false,
        ″valType″: ″date″,
        ″dataFormat″: ″1900-01-01″,
        ″primaryKey″: false
      },
      {
        ″name″: ″Id″,
        ″comment″: ″Unique identifier″,
        ″user″: ″A″,
        ″visualizeOnMap″: false,
        ″visualizeOnTable″: true,
        ″valType″: ″string″,
        ″primaryKey″: true
      },
      ....
      {
        ″name″: ″Latitude″,
        ″comment″: ″Latitude″,
        ″user″: ″A″,
        ″visualizeOnMap″: false,
        ″visualizeOnTable″: true,
        ″valType″: ″geographicCoordinate″,
        ″primaryKey″: false
      },
      {
        ″name″: ″Longitude″,
        ″comment″: ″Longitude″,
        ″user″: ″A″,
        ″visualizeOnMap″: false,
        ″visualizeOnTable″: true,
        ″valType″: ″geographicCoordinate″,
        ″primaryKey″: false
      },
      …
]
```

RECORD SECTION

The record section defines the values for each point (according to the “field” defined in the previous scheme).

```
″record″: [
       {
        ″Table″: ″DatiCraftBusinesses″,
        ″Date″: ″2021-02-01″,
        ″Id″: ″1″,
        ...
        ″Latitude″: 41.18,
        ″Longitude″: 16.21,
        ...
      },
      {
        ″Table″: ″DatiCraftBusinesses″,
        ″Date″: ″2021-02-01″,
        ″Id″: ″2″,
        ...
```

```
            ″Latitude″: 41.184,
            ″Longitude″: 16.33,
            . . .

        }
        …
]
```

In this case, the system will present the possibility to choose the typology of tangible and intangible heritage or CCIs to be visualized on the map.

Once the user chooses this typology, the craft activities and their characteristics will be automatically loaded and displayed on the screen, depending on the definition and permissions, as the point of interest on the urban space.

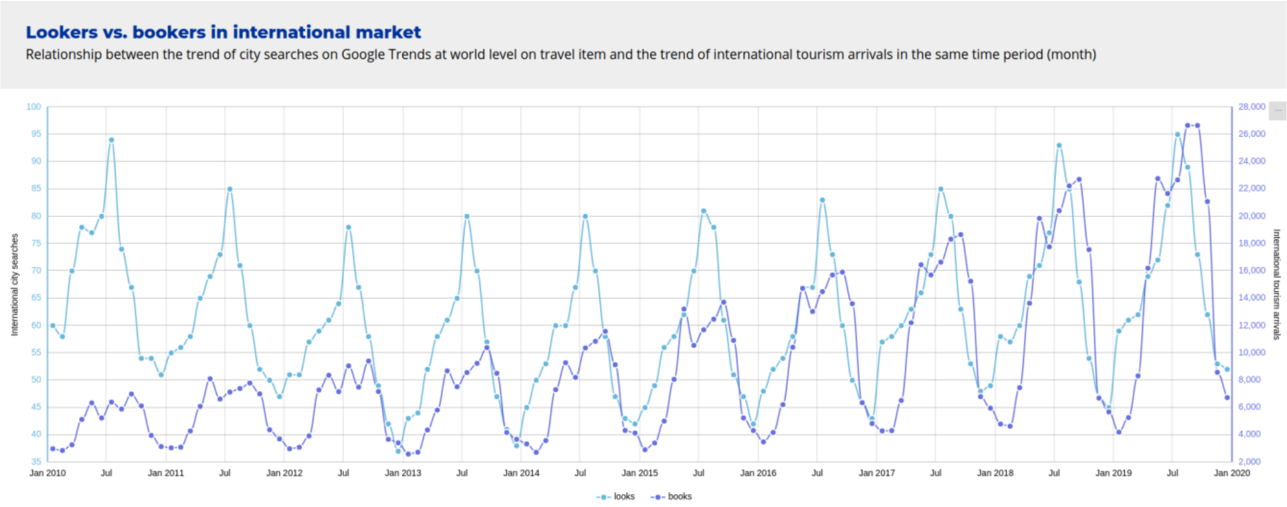
Dynamic maps allow the user to:

- **displays** each of the identified handcraft and CCI Point of Interest - POI (artisans, craft and CCI businesses and organizations) **with a marker placed** on the map of the city,
- **visualizes additional information** about this POI (for example the name, the kind of product/activities, kind of activities/experiences offered to visitors, etc.), by clicking on the marker,
- additional information according to the criteria identified in D.3.3.3. The dynamic map displays only the POIs that respond to the selected parameters.

4. Quering the Datahub

From the datahub to the dashboard: how to show the data

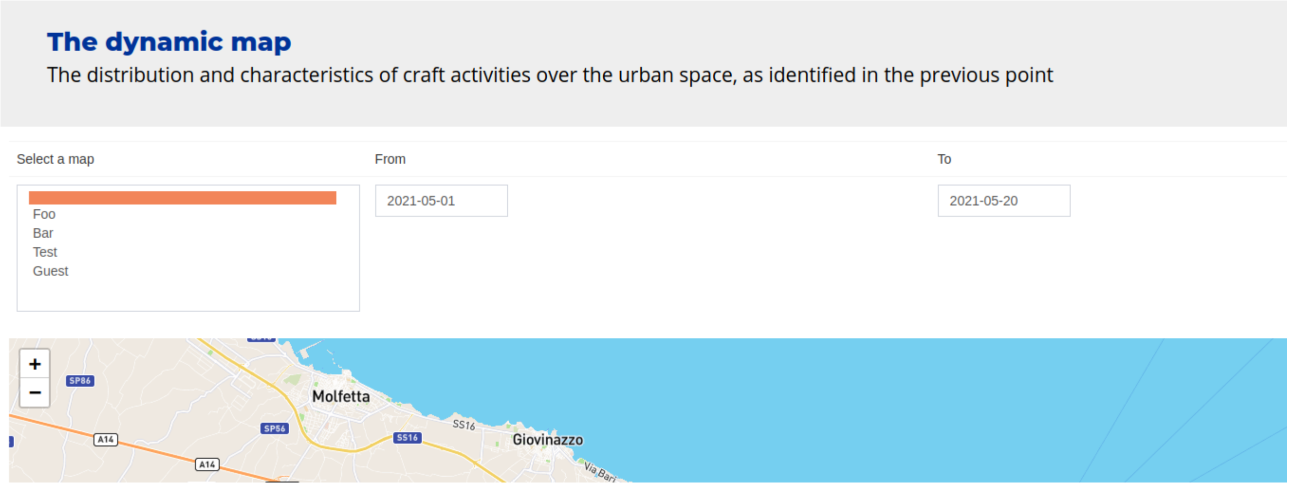
Each section within the dashboard has a preset of data already filtered by topic and sub-topic of the existing data. Clicking on the corresponding menu item the user can simply access the data. Depending on the view provided, it is possible to filter further on a time basis [graphic zoom].



Lookers vs. bookers in international market

Relationship between the trend of city searches on Google Trends at world level on travel item and the trend of international tourism arrivals in the same time period (month)

Figure 8. Example of data standard viewed in the dashboard



The dynamic map

The distribution and characteristics of craft activities over the urban space, as identified in the previous point

Select a map

Foo
Bar
Test
Guest

From: 2021-05-01

To: 2021-05-20

Figure 9. Example of time intervals [from / to] in the dashboard

It is not possible, within the dashboard, to perform free search queries.

There is the possibility, upon request, to have access to:

1) the generic dataset of public data for all cities,
2) the specific dataset of data relating to the specific city.

4.1. Public data

Public data are related to tourism and non-tourism performances from different urban perspectives (economic, social, environmental, cultural, mobility, etc.).

Each data is linked to the reference destination through the city filter, as follows:

```
city:″[city name]″


   city: ″Bari″,
   city: ″Duborvnik″,
   city: ″Ferrara″,
   city: ″Sibenik″,
   city: ″Venezia″,
```

In addition, each record has two additional filters: kind and file.

KIND: “kind” (for the Italian language) or “kind_en” (for the English language) allow the user to filter values for a generic data type (for example kind_en: "Population")

FILE: file allows the user to filter the values based on the source name from which they were imported (*e.g. file: "P1" gives you “Population” exactly as kind_en: ”Population*”).

e.g.: obtain data on the population of Venice

```
find({city:″Venezia″,kind_en:″Population″})
```

or

```
find({city: ″Venezia″,file:″P1″})
```

gives you the same result

The complete “topic”→”File” map will be provided only on request because each city is mapped differently.

Google Trends

This section represents all data imports related to google trends.
It is possible to filter the data of a specific destination, based on the filter: `city: “[name]”`

Furthermore, each record is represented by:

- year,
- month,
- categories [can be “All”,”Travel”,”Viaggi”,”Ait Travel”,”Hotels and accommodations”],
- area [can be “All”,”Italy”,”Croatia”].

Feeds

All the imports are made automatically by the RSS feeds relating to news and weather.
It is possible to filter the data of a specific city based on the filter:

- city:”[name]”

Further, each record is represented by:

- kind [can be “news”,” weather”]
- day [format: yyyy-mm-dd]

4.2. Private data

The datahub also contains data originating from sources that do not allow them to be made public and shown on the datahub.

These sources are:

- destination cameras and sensors: these tools detect sensitive data belonging to natural persons. These data are never collected in the datahub,
- Tripadvisor: data from Tripadvisor were linked to the temporality of the contract stipulated between the company and Ca’ Foscari University (LP) it possible to visualize on the dashboard just the subset,
- personal data of craft business owners collected within WP 3.3.,
- personal data of destination stakeholders (users defined by the destination itself).

It is important to underline that every dashboard and its data are accessible only by the stakeholder authorized by the city partner.

4.3. Datahub query details

4.3.1. City at a Glance

City population

The trend of population in the Municipality and the historical Center

```
source: NoSQL
collection: Datas
query [ all cities]:
        db.Datas.find({“city”:”[cityName]”, file:”P1”})
        reply:
        {
                "_id" : ObjectId("60648bfdb105f729c67f97dc"),
                "kind" : "Popolazione",
                "kind_en" : "Population",
                "subkind" : "Dati socio-demografici",
                "subkind_en" : "Socio-demographic data",
                "indicator" : "City at a glance",
                "file" : "P1",
                "city" : "[cityName]",
                "values" : [
                        {
                                "anno" : 2007,
                                "qty" : 133214
                        },
                        {
                                "anno" : 2008,
                                "qty" : 133591
                        } …
                        {
                                "anno" : 2018,
                                "qty" : 132278
                        }
                ]
        }
```

Vibrancy of city population

Evolution of Millennials population and Senior population

```
source: NoSql
collection: Datas
query: [ all cities ]
        db.Datas.find({“city”:”[cityName]”,            file:”P2”})            +
        db.Datas.find({“city”:”[cityName]”, file:”P1”})
        reply:
                p1                                                  reply
                +
                {
                        "_id" : ObjectId("602fa22fc035064066597a74"),
"kind" : "Popolazione",
"kind_en" : "Population",
"subkind" : "Dati socio-demografici",
"subkind_en" : "Socio-demographic data",
"city" : "[cityName]",
"question" : "classe d'eta",
"question_en" : "age range",
"indicator" : "City at a glance",
"file" : "P2",
"values" : [
{
        "anno" : 2010,
        "classe d'eta" : "0-4",
        "centro" : 7286,
        "comune" : 10405
},
{
        "anno" : 2010,
        "classe d'eta" : "5-9",
        "centro" : 7389,
        "comune" : 10871
},

...
{
        "anno" : 2010,
        "classe d'eta" : "90+",
        "centro" : 7389,
        "comune" : 10871
}
…
{
        "anno" : 2018,
        "classe d'eta" : "90+",
                        "centro" : 1087,
                        "comune" : 4653
                    }]
                }
```

NOTE: age range may differ from city to city.

Openness to the world

Evolution of population foreign-born

```
        source: NoSql
        collection: Datas
        query: [ IT cities ]
            db.Datas.find({“city”:”[cityName]”, file:”P6”}) +
            db.Datas.find({“city”:”[cityName]”, file:”P1”})
        reply:
                p1 reply
                +
                {
                    "_id" : ObjectId("5eccf0f9fe0295121b57d8a4"),
                    "kind" : "Popolazione residente con cittadinanza estera",
                    "kind_en" : "Population that is foreign born",
                    "subkind" : "Dati socio-demografici",
                    "subkind_en" : "Socio-demographic data",
                    "city" : "[cityName]",
                    "indicator" : "City at a glance",
                    "file" : "P6"
                    "values" : [
                        {
                            "anno" : 2008,
                            "qty" : 6938
                        },
                        …
                        {
                            "anno" : 2019,
                            "qty" : 14167
                        }
                    ]
                }
        query: [ Dubrovnik ]
            db.Datas.find({“city”:”[cityName]”,          file:”P5”})          +
            db.Datas.find({“city”:”[cityName]”, file:”P1”})
        reply:
                p1                                                  reply
                +
                {
                    "_id" : ObjectId("5eccf0f9fe0295121b57d8a4"),
                    "kind" : "Popolazione residente con cittadinanza estera",
                    "kind_en" : "Population that is foreign born",
                    "subkind" : "Dati socio-demografici",
                    "subkind_en" : "Socio-demographic data",
                    "city" : "[cityName]",
                    "indicator" : "City at a glance",
                    "file" : "P5"
                    "values" : [
                        {
                            "anno" : 2008,
                            "qty" : 6938
                        },
                        …
                        {
                            "anno" : 2019,
                            "qty" : 14167
                        }
                    ]
                }
```

Economic wealth of the population

Evolution of per capita personal income before taxes

```
source: NoSql
collection: Datas
query: [ All cities ]
        db.Datas.find({“city”:”[cityName]”, file:”E1”})
reply:
        differs city by city
        the main structure is like
        {
                "_id" : ObjectId("5e9f1c4a238e571ab2144673"),
                "kind" : "Redditi IRPEF",
                "kind_en" : "GDP per capita",
                "city" : "[cityName]",
                "indicator" : "City at a glance",
                "file" : "E1",
                "values" : [
                        {
                                "anno" : 2010,
                                "qty" : 10155.548565306097
                        }
                ]
        }
```

City employment rate

Evolution of employment rate

```
source: NoSql
collection: Datas
query: [ IT cities ]
        db.Datas.find({“city”:”[cityName]”, file:”P2”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E3”})
reply:
        p2 reply +
        {
                "_id" : ObjectId("5e8f76331686f359813772d9"),
                "kind" : "No. addetti totali",
                "kind_en" : "No. of total people employed",
                "city" : "[cityName]",
                "indicator" : "City at a glance",
                "file" : "E3",
                "values" : [
                        {
                                "anno" : 2012,
                                "qty" : 40726.69
                        },
                ]
        }
query: [ HR cities ]
        db.Datas.find({“city”:”[cityName]”, file:”P1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E4”})
reply:
        p1 reply +
        {
                "_id" : ObjectId("5e8f1a48da078722797ceb40"),
                "kind" : "City employment",
                "kind_en" : "City employment",
                "city" : "[cityName]",
                "indicator" : "City at a glance",
                "file" : "E4",
                "values" : [
                        {
                                "anno" : 2011,
                                "qty" : 12574
                        },

                ]
        }
```

Specialization in high knowledge intensity services (HIKS)

Evolution of employment in high knowledge intensity services

```
source: NoSql
collection: Datas
query: [ Bari|Ferrara ]
        db.Datas.find({“city”:”[cityName]”, file:”P1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E4”})
reply:
        P1 reply + E4 reply

query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”E3”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E4”})
reply:
        E3 reply + E4 reply
                query [ HR cities ]
        db.Datas.find({“city”:”[cityName]”, file:”E4”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E6”})
reply: E4 reply +
        {
                "_id" : ObjectId("5e8f1a4ada078722797ceb41"),
                "kind" : "No. addetti in settori ad alta intensità di conoscenza",
                "kind_en" : "No.of people employed in high knowledge intensity sectors",
                "city" : “[cityName]",
                "indicator" : "City at a glance",
                "file" : "E6",
                "values" : [
                        {
                                "anno" : 2007,
                                "qty" : 666
                        }
                ]
        }
```

Tourism specialization of entrepreneurship

Evolution of tourism businesses

```
source: NoSql
collection: Datas
query: [ Bari ]
        db.Datas.find({“city”:”[cityName]”, file:”P1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E4”})
reply:
        P1 reply + E4 reply

query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”E3”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E4”})
reply:
        E3 reply + E4 reply
                query [ HR cities ]
        db.Datas.find({“city”:”[cityName]”, file:”E4”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E6”})
reply: E4 reply +
        {
                "_id" : ObjectId("5e8f1a4ada078722797ceb41"),
                "kind" : "No. addetti in settori ad alta intensità di conoscenza",
                "kind_en" : "No.of people employed in high knowledge intensity sectors",
                "city" : “[cityName]",
                "indicator" : "City at a glance",
                "file" : "E6",
                "values" : [
                        {
                                "anno" : 2007,
                                "qty" : 666
                        }
                ]
        }
```

Direct employment in tourism

Evolution of employment in tourism

```
source: NoSql
collection: Datas
query: [ IT cities ]
        db.Datas.find({“city”:”[cityName]”, file:”E3”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E7”})
reply:
        E3 reply +
        {
                "_id" : ObjectId("5e904a2a1686f359813772dc"),
                "kind" : "No di addetti nei settori dell'ospitalità, della ristorazione, della ristorazione e dell'intermediazione",
                "kind_en" : "No. people employed in tourism",
                "city" : "[cityName]",
                "indicator" : "tourism",
                "file" : "E7",
                "values" : [
                        {
                                "anno" : "2012",
                                "qty" : 7018.54
                        }
                ]
        }
query: [ HR cities ]
        db.Datas.find({“city”:”[cityName]”, file:”E4”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T26”})
reply:
```

```
E4 reply +
{
    "_id" : ObjectId("5e8f189433513a65445c8e24"),
    "kind" : "No di impiegati nel settore del turismo",
    "kind_en" : "No of people employed in tourism businesses",
    "city" : "Sibenik",
    "indicator" : "tourism",
    "file" : "T26",
    "values" : [
        {
            "anno" : 2007,
            "qty" : 927
        }
    ]
}
```

Tourism specialization of entrepreneurship

Evolution of tourism businesses

```
source: NoSql
collection: Datas
query: [ IT cities ]
    db.Datas.find({“city”:”[cityName]”, file:”E2”}) +
    db.Datas.find({“city”:”[cityName]”, file:”E6”})
reply:
    E2 reply + E6 reply

query: [ HR cities ]
    db.Datas.find({“city”:”[cityName]”, file:”E5”}) +
    db.Datas.find({“city”:”[cityName]”, file:”T25”})
reply:
    E6 reply +
    {
        "_id" : ObjectId("606d8f0c860dc63de66659a7"),
        "kind" : "No di attività nei settori dell'ospitalità, della ristorazione e dell'intermediazione",
        "kind_en" : "No. of businesses in hospitality, catering and intermediation",
        "city" : "Sibenik",
        "indicator" : "tourism",
        "file" : "T25",
        "values" : [
            {
                "anno" : 2007,
                "qty" : 78
            }
        ]
    }
```

Tourism market vs. demographic development

Correlation between the evolution of tourism arrivals vs. city population

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari|HR cities ]
    db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
    db.Datas.find({“city”:”[cityName]”, file:”P1”})
reply:
    P1 reply +
    {
        "_id" : ObjectId("60269405ae1f84071d4d28d5"),
        "kind" : "Arrivi",
        "kind_en" : "Total tourism arrivals in the city",
        "indicator" : "tourism",
        "file" : "T1",
        "city" : "Sibenik",
        "values" : [
            {
                "anno" : 2010,
                "comune" : 194782,
                "qty" : 194782
            }
        ]
    }
query: [ Venezia ]
    db.Datas.find({“city”:”[cityName]”, file:”P1”}) +
    db.Datas.find({“city”:”[cityName]”, file:”A9”})
reply:
    P1 reply +
    {
        "_id" : ObjectId("6022907fd1029d485c19cec7"),
        "kind" : "Arrivi",
        "kind_en" : "Total tourism arrivals in the city",
        "city" : "Venezia",
        "file" : "A9",
        "indicator" : "tourism",
        "values" : [
            {
                "anno" : 2010,
                "centro" : 2251160,
                "comune" : 3708407
            }
        ]
    }
```

Tourism market vs. hotel accommodation development

Correlations between the evolution of tourism arrivals vs. hotel bed places

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
```

```
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T17”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A24”})
query: [ Sibenik ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T18”})
query: [ Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T13”})
reply:
        T1 or A9 reply +
        [ A24 ]
        {
                "kind" : "Posti letto strutture alberghiere",
                "kind_en" : "No. of hotel bedplaces",
                "file" : "A24",
                "indicator" : "tourism",
                "city" : "Venezia",
                "values" : [
                        {
                                "anno" : 2010,
                                "centro" : 16258,
                                "comune" : 28098
                        }
                ]
        }
        [ T17 ]
        {
                "kind" : "Posti letto strutture alberghiere",
                "kind_en" : "No. of hotel bedplaces",
                "city" : "Bari",
                "indicator" : "tourism",
                "file" : "T17",
                "values" : [
                        {
                                "anno" : 2012,
                                "kind" : " » alberghi 5 stelle lusso",
                                "qty" : 153

                ]
        }
        [T13|T18]
        {
                "kind" : "Posti letto strutture alberghiere",
                "kind_en" : "No. of hotel bedplaces",
                "city" : "Sibenik",
                "indicator" : "tourism",
                "file" : "T18",
                "values" : [
                        {
                                "anno" : 2010,
                                "qty" : 3048
                        }
                ]
        }
```

Tourism market vs. non-hotel accommodation development

Correlations between the evolution of tourism arrivals vs. non-hotel bed places.

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T19”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A26”})
```

```
query: [ Sibenik ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T20”})
query: [ Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T15”})
reply: [ A26 ]
        {
                "kind" : "Posti letto strutture non alberghiere",
                "kind_en" : "No. of non- hotel bedplaces",
                "city" : "Venezia",
                "file" : "A26",
                "indicator" : "tourism",
                "values" : [
                        {
                                "anno" : 2010,
                                "comune" : 17345,
                                "centro" : 11582
                        }
                ]
        }
        [T19]
        {
                "kind" : "Posti letto strutture non alberghiere",
                "kind_en" : "No. of non- hotel bedplaces by typology (registered data)",
                "city" : "Bari",
                "indicator" : "tourism",
                "file" : "T19",
                "values" : [
                        {
                                "anno" : 2010,
                                "kind" : " » affittacamere",
                                "qty" : 38
                        }
                ]
        }
        [T20|T15]
        {
                "kind" : "Posti letto strutture non alberghiere",
                "kind_en" : "Non hotel bedplaces",
                "city" : "Sibenik",
                "indicator" : "tourism",
                "file" : "T20",
                "values" : [
                        {
                                "anno" : 2010,
                                "qty" : 5357
                        }
                ]
        }
```

Tourism market vs. business development

Correlations between the evolution of tourism arrivals vs. tourism businesses

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E6”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E6”})
query: [ HRcities ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T25”})

reply: T1 or A9  +
      E6 or T25
```

Tourism market vs. employment in tourism

Correlations between the evolution of tourism arrivals vs. tourism businesses

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E3”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E3”})
query: [ HRcities ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T26”})

reply: T1 or A9  +
      E3 or T26
```

4.3.2. Tourism

How many tourists in my city?

Trend in total arrivals/nights and evolution of total tourism arrivals and nights

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari|HRcities ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T2”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A10”})
```

reply: T1 or A9 +

```
[A10]
{
        "_id" : ObjectId("6022922263dbd175b133eee9"),
        "kind" : "Pernottamenti",
        "kind_en" : "Total tourism nights in the city",
        "city" : "Venezia",
        "file" : "A10",
        "indicator" : "tourism",
        "values" : [
                {
                        "anno" : 2010,
                        "comune" : 8521247,
                        "centro" : 5760811
                }
        ]
or
[T2]
{
        _id" : ObjectId("6026860fb6816652b84f8040"),
        "kind" : "Pernottamenti",
        "kind_en" : "Total tourism nights in the city",
        "city" : "Bari",
        "indicator" : "tourism",
        "file" : "T2",
        "values" : [
                {
                        "anno" : 2010,
                        "comune" : 514885,
                        "qty" : 514885
                }
        ]
    }
```

Trends in international tourism

Trends in international tourism arrivals and nights: international arrivals and nights; % arrivals and overnights stays

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari|HRcities ]
        db.Datas.find({“city”:”[cityName]”, file:”T3”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T4”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A11”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A12”})

reply:
        [T3]
                {
                        "kind" : "Arrivi internazionali",
                        "kind_en" : "Total international tourism arrivals in the city",
                        "city" : "Bari",
                        "indicator" : "tourism",
                        "file" : "T3",
                        "values" : [
                                {
                                        "anno" : 2010,
                                        "comune" : 58330,
                                        "qty" : 58330
                                }
                        ]
                }
        or [A11]
                {
                        "kind" : "Arrivi internazionali",
                        "kind_en" : "Total international tourism arrivals in the city",
                        "file" : "A11",
                        "city" : "Venezia",
                        "indicator" : "tourism",
                        "values" : [
                                {
                                        "anno" : 2010,
                                        "comune" : 3107956,
                                        "centro" : 1956957
                                }
                        ]
                }
        +
        [T4]
                {
                        "kind" : "Pernottamenti internazionali",
                        "kind_en" : "Total international tourism nights in the city",
                        "city" : "Sibenik",
                        "indicator" : "Tourism",
                        "file" : "T4",
                        "values" : [
                                {
                                        "anno" : 2010,
                                        "qty" : 122933
                                }
                        ]
                }
```

```
or [ A12]
    {
        "kind" : "Pernottamenti internazionali",
        "kind_en" : "Total international tourism nights in the city",
        "city" : "Venezia",
        "file" : "A12",
        "indicator" : "tourism",
        "values" : [
            {
                "anno" : 2010,
                "centro" : 4988678,
                "comune" : 7064202
            }
        ]
    }
```

Trends in domestic tourism

Trends in domestic arrivals and nights: **domestic arrivals and nights; % arrivals and overnights stays**

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari|HRCities ]
    db.Datas.find({“city”:”[cityName]”, file:”T5”}) +
    db.Datas.find({“city”:”[cityName]”, file:”T6”})
query: [ Venezia ]
    db.Datas.find({“city”:”[cityName]”, file:”A13”}) +
    db.Datas.find({“city”:”[cityName]”, file:”A14”})
reply:
    [T5]
        {
            "_id" : ObjectId("6026928f3a4fea4b913b77b3"),
            "kind" : "Arrivi naziuonali",
            "kind_en" : "Total domestic tourism arrivals in the city",
            "city" : "Sibenik",
            "indicator" : "tourism",
            "file" : "T5",
            "values" : [
                {
                    "anno" : 2010,
                    "qty" : 26139
                }
            ]
        }
```

or [A13]

```
        {
            "_id" : ObjectId("6022924a8b19b22a097caec7"),
            "kind" : "Arrivi nazionali",
            "kind_en" : "Total domestic tourism arrivals in the city",
            "city" : "Venezia",
            "file" : "A13",
            "indicator" : "tourism",
            "values" : [
                {
                    "anno" : 2010,
                    "centro" : 294203,
                    "comune" : 600451
                }
            ]
        }
```

+ [T6]

```
        {
            "_id" : ObjectId("6026928c3a4fea4b913b77b2"),
            "kind" : "Pernottamenti nazionali",
            "kind_en" : "Total domestic tourism nights in the city",
            "city" : "Sibenik",
                "indicator" : "tourism",
                "file" : "T6",
                "values" : [
                        {
                                "anno" : 2010,
                                "qty" : 104626
                        }
                ]
        }
or [A14]
        {
                "_id" : ObjectId("602292734ad7b3658514d7c6"),
                "kind" : "Pernottamenti nazionali",
                "kind_en" : "Total domestic tourism nights in the city",
                "city" : "Venezia",
                "file" : "A14",
                "indicator" : "tourism",
                "values" : [
                        {
                        "anno" : 2010,
                        "centro" : 722133,
                        "comune" : 1457045
                        }
                ]
        }
```

Internationalization of tourism

Importance of international tourism for the city: % share of international arrivals on total arrivals/years

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari|HRCities ]
        db.Datas.find({“city”:”[cityName]”, file:”T3”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T1”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A11”})
reply: T3 or A9  + T1 or A11
```

Who are my international clients?

Concentration of international arrivals from the first 5 countries of origin: internationalization concentration/years

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari|HRCities ]
        db.Datas.find({“city”:”[cityName]”, file:”T3”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T7”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A11”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A16bis”})
reply: T3 or A11 +
        [T7] or [A16bis ]
                {
                        "_id" : ObjectId("607eb62efcfa5a556f3a0887"),
                        "kind" : "Arrivi internazionali per paese",
                        "kind_en" : "Total international tourism arrivals by country of origin",
                        "city" : "Dubrovnik",
                        "indicator" : "tourism",
                        "file" : "T7 | A16bis",
                        "values" : [
                                {
                                        "anno" : 2010,
                                        "nazione" : "united kingdom",
                                        "qty" : 66197
                                }
```

```
            ]
        }
```

Who are my domestic clients?

Concentration of domestic arrivals from the first 5 regions of origin: domestic concentration/years

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
    db.Datas.find({“city”:”[cityName]”, file:”T5”}) +
    db.Datas.find({“city”:”[cityName]”, file:”T9”})
query: [ Venezia ]
    db.Datas.find({“city”:”[cityName]”, file:”A13”}) +
    db.Datas.find({“city”:”[cityName]”, file:”A17bis”})
reply: T5 or A13
    + [T9] or [A17bis]
        {
            "_id" : ObjectId("60268c6f4b8fff3eb74c4ea7"),
            "kind" : "Arrivi nazionali per regione",
            "kind_en" : "Total domestic tourism arrivals by region of origin",
            "city" : "Ferrara",
            "indicator" : "tourism",
            "file" : "T9|A17bis",
            "values" : [
                {
                    "anno" : 2007,
                    "regione" : "Piemonte",
                    "qty" : 11806
                }
            ]
        }
```

Seasonality of international tourism

Monthly trend in international arrivals and concentration of international tourists in high season

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
    db.Datas.find({“city”:”[cityName]”, file:”T11”}) +
    db.Datas.find({“city”:”[cityName]”, file:”T3”})
query: [ Venezia ]
    db.Datas.find({“city”:”[cityName]”, file:”A18”}) +
    db.Datas.find({“city”:”[cityName]”, file:”A11”})
query: [ HRcities ]
    db.Datas.find({“city”:”[cityName]”, file:”T9”}) +
    db.Datas.find({“city”:”[cityName]”, file:”T3”})

reply:
    [T11] or [A18] or [T8]
        {
            "_id" : ObjectId("607f04d468d9e56ccd710ce9"),
            "kind" : "Arrivi internazionali per mese",
            "kind_en" : "International arrivals by month",
            "city" : "Ferrara",
            "indicator" : "tourism",
            "file" : "T11 | A18 | T8",
            "values" : [
                {
                    "anno" : 2007,
                    "mese" : "January",
                    "mese_int" : 1,
                    "qty" : 1889
                }
            ]
        }
    +
```

T3 or A11

Seasonality of domestic tourism

Monthly trend in domestic arrivals and concentration of domestic tourists in high season

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T13”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T5”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A19bis”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A13”})
query: [ HRCities ]
        db.Datas.find({“city”:”[cityName]”, file:”T9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T5”})
query: [ Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”T9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T5”})

reply:



        [T13 | A19bis | T9 ]
                {
                        "_id" : ObjectId("60268e23b6816652b84f8125"),
                        "kind" : "Arrivi nazionali per mese",
                        "kind_en" : "Domestic arrivals by month",
                        "city" : "Ferrara",
                        "indicator" : "tourism",
                        "file" : "T13 | A19bis",
                        "values" : [
                {
                    "anno" : 2007,
                    "mese_int" : 1,
                    "mese" : "January",
                    "qty" : 8386
            ]
        }
    +
    T5 or A13
```

Cruise tourism

Trends in cruise passengers

```
source: NoSql
collection: Datas
query: [ Bari| ] db.Datas.find({“city”:”[cityName]”, file:”T20”})
query: [ Venezia ] db.Datas.find({“city”:”[cityName]”, file:”A21”})
query: [ Sibenik ] db.Datas.find({“city”:”[cityName]”, file:”T12”})
query: [ Dubrovnik ] db.Datas.find({“city”:”[cityName]”, file:”T16”})
reply:
    [T20 | A21 ]
        {
            "_id" : ObjectId("607dae5ca38aa94bab0d20b4"),
            "kind" : "Passeggeri di crociere",
            "kind_en" : "No. of passengers on cruises",
            "city" : "[cityName]",
            "indicator" : "tourism",
            "file" : "T20 | A21",
            "values" : [
                { "anno" : 2009, "home" : 0, "transit" : 0, "qty" : 567692 }
            ]
        }
    [T12]
```

```
        {
                "_id" : ObjectId("6026931eb6816652b84f816a"),
                "kind" : "Passeggeri di crociere",
                "kind_en" : "No. of passengers on cruises",
                "indicator" : "tourism",
                "file" : "T12",
                "city" : "Sibenik",
                "values" : [
                        { “anno”: 2006, "qty" : 4836 }
                ]
        }
```

[T16]

```
        {
                "_id" : ObjectId("5eb2c11609fa140dc77249b5"),
        "kind" : "Passeggeri di crociere",
                "kind_en" : "No. of passengers on cruises",
                "city" : "Dubrovnik",
                "indicator" : "tourism",
                "file" : "T16",
                "values" : [
                        {
                                "anno" : 2020, "mese" : 1,
                                "port" : 2880, "center" : null, "qty" : 2880
                        }
                ]
        }
```

Tourist Pressure

Average daily tourist presences every 100 inhabitants: ratio of daily tourist presence every 100 inhabitants/years

```
source: NoSql
collection: Datas
query: [ Bari|Ferrara|HRcities ]
        db.Datas.find({“city”:”[cityName]”, file:”T2}) +
        db.Datas.find({“city”:”[cityName]”, file:”P1}})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A10”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A11”}) +
        db.Datas.find({“city”:”[cityName]”, file:”P1”})

reply: T2 or ( A10 & A11 )  +  P1
```

Official accommodation supply – Establishments

Trend in the hotel and non-hotel establishments: number of establishments/years Evolution of hotel and non-hotel establishments annual % change/years

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T16”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T17”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T18”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T19”}) +
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A23”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A24”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A25”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A26”})
query: [ Sibenik ]
        db.Datas.find({“city”:”[cityName]”, file:”T17”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T18”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T19”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T20”})
quert: [ Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”T12”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T13”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T14”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T25”})




reply:
        [ Bari: T16 | T17 | T18 | T19 ]
            {
                "kind" : "Strutture alberghiere",
                "kind_en" : "No. of hotel establishments",
                "city" : "Bari",
                "indicator" : "tourism",
                "file" : "T16 | T17 | T18 | 19",
                "values" : [
                    { "anno" : 2010, "kind" : " » alberghi 5 stelle lusso", "qty" : 0 }
                ]
            }
        [ Venezia: A23 | A24 | A25 | A26]
            {
                "kind" : "Strutture alberghiere",
                "kind_en" : "No. of hotel establishments",
                "city" : "Venezia",
                "file" : "A23 | A24 | A25 | A26",
                "indicator" : "tourism",
                "values" : [
                        { "anno" : 2010, "centro" : 251, "comune" : 379 }
                ]
        }
[ Sibenik: T17 | T18Sibenik | T19Sibenik | T20Sibenik ]
        {
                "kind" : "Strutture alberghiere",
                "kind_en" : "No. of hotel establishments",
                "city" : "Venezia",
                "file" : "T17 | T18 | T19 | T20",
                "indicator" : "tourism",
                "values" : [
                        { "anno" : 2010, "centro" : 251, "comune" : 379 }
                ]
        }
[ Dubrovnik: T12 | T13 | T14 | T25 ]
        {
                "kind" : "Strutture alberghiere",
                "kind_en" : "Hotel establishment",
                "city" : "Dubrovnik",
                "indicator" : "tourism", "file" : "T12 | T13 | T14 | T25",
                "values" : [ { "anno" : 2016, "qty" : 48 }]
        }
```

Official accommodation supply – Bed places

Trend in a hotel and non-hotel bed places: number of bed places/years
Evolution of hotel and non-hotel bed places: annual % change/years

source: NoSql
collection: Datas
query: [Ferrara|Bari|]

```
        db.Datas.find({“city”:”[cityName]”, file:”T16”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T17”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T18”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T19”}) +
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A23”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A24”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A25”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A26”})
query: [ Sibenik ]
        db.Datas.find({“city”:”[cityName]”, file:”T17”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T18”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T19”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T20”})
quert: [ Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”T12”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T13”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T14”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T25”})
reply:
        [ Bari: T16 | T17 | T18 | T19 ]
                {
                        "kind" : "Strutture alberghiere",
                        "kind_en" : "No. of hotel establishments",
                        "city" : "Bari",
                        "indicator" : "tourism",
                        "file" : "T16 | T17 | T18 | 19",
                        "values" : [
                                { "anno" : 2010, "kind" : " » alberghi 5 stelle lusso", "qty" : 0 }
                        ]
                }
        [ Venezia: A23 | A24 | A25 | A26]
                {
                        "kind" : "Strutture alberghiere",
                        "kind_en" : "No. of hotel establishments",
                        "city" : "Venezia",
                        "file" : "A23 | A24 | A25 | A26",
                        "indicator" : "tourism",
                        "values" : [
                                { "anno" : 2010, "centro" : 251, "comune" : 379 }
                        ]
                }
        [ Sibenik: T17 | T18Sibenik | T19Sibenik | T20Sibenik ]
                {
                        "kind" : "Strutture alberghiere",
                        "kind_en" : "No. of hotel establishments",
                        "city" : "Venezia",
                        "file" : "T17 | T18 | T19 | T20",
                        "indicator" : "tourism",
                        "values" : [
                                { "anno" : 2010, "centro" : 251, "comune" : 379 }
                        ]
                }




        [ Dubrovnik: T12 | T13 | T14 | T25 ]
                {
                        "kind" : "Strutture alberghiere",
                        "kind_en" : "Hotel establishment",
                        "city" : "Dubrovnik",
                        "indicator" : "tourism", "file" : "T12 | T13 | T14 | T25",
                        "values" : [ { "anno" : 2016, "qty" : 48 }]
                }
```

Role of hotel supply

Importance of hotels on the city accommodation supply: % hotel bed places on total bed places/years

```
source: NoSql
collection: Datas
query: [ Ferrara|Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T17”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T19”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A24”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A26”})
query: [ Sibenik ]
        db.Datas.find({“city”:”[cityName]”, file:”T18”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T20”})
query: [ Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”T13”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T15”})
reply:
        Bari | Ferrara T17 + T19
        Venezia: A24+ A26
        Sibenik: T18 + T20
        Dubrovnik: T13 + T15
```

Tourism market vs. air accessibility

Evolution of tourism arrivals vs. air accessibility

```
source: NoSql
collection: Datas
query: [ Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T11”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T13”})
query: [ Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”T8”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T9”})
```

```
reply: T8 or T11 + T13 or T9
```

Tourism market vs. cruise accessibility

Evolution of tourism arrivals vs. cruise accessibility

```
source: NoSql
collection: Datas
query: [ Bari| ]
        db.Datas.find({“city”:”[cityName]”, file:”T1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T20”})
query: [ Venezia ]
        db.Datas.find({“city”:”[cityName]”, file:”A9”}) +
        db.Datas.find({“city”:”[cityName]”, file:”A21”})
query: [ HRcities ]
        db.Datas.find({“city”:”[cityName]”, file:”T9bis”}) +
        db.Datas.find({“city”:”[cityName]”, file:”T12”})

reply:
        T1 or A9 or
        [ T9bis ]
                {
                        "kind" : "Totale arrivi per mese",
                        "kind_en" : "Total arrivals by month",
                        "city" : "Sibenik",
                        "section" : "tourism",
                        "file" : "T9bis",
                        "values" : [
                                { "anno" : 2010, "mese" : "Jan", "mese_int" : 1, "qty" : 478 }
                        ]
                }
        + T20 or A21 or T12
```

4.3.3. Culture and Craft

Employment in culture, arts and entertainment activities

Number of people employed in those sectors on total employed: % share on total employed / years

```
source: NoSql
collection: Datas
query: [ ITCities ]
        db.Datas.find({“city”:”[cityName]”, file:”E3”}) +
        db.Datas.find({“city”:”[cityName]”, file:”E8”})
reply:
        E3                                                                    +
        [E8 ]
        {
                "kind" : "No addetti nei settori dell'arte, della cultura e dell'intrattenimento",
                "kind_en" : "No. of people employed in art and culture",
                "city" : "Bari",
                "indicator" : "City ad a glance",
                "file" : "E8",
                "values" : [ { "anno" : 2012, "qty" : 261.63 }]
        }
```

The craft activities identified by SLIDES

List of craft activities by type of production

```
source: NoSql
collection: Datas
query: [ All Cities ]
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori1”})
reply:
        {
                "kind" : "The proper craft activities identified by the SLIDES project by type of
                        production",
                "kind_en" : "The proper craft activities identified by the SLIDES project by
                        type of production",
                "file" : "Indicatori1",
                "city" : [cityName]",
                "indicator" : "Culture and craft",
                "values" : [
                        {
                            "anno" : 2020,
                            "type" : "Local Food and beverages",
                            "abs" : 177,
                            "perc" : 86.3
                        }
                ]
        }
```

The main features of craftsmen

Contemporary craft / Contemporary folk arts (combination between tradition and innovative technologies, materials on arts vs. traditional craft/folk customs and arts)

```
source: NoSql
collection: Datas
query: [ All Cities ]
    db.Datas.find({“city”:”[cityName]”, file:”Indicatori2”})
reply:
    {
         "_id" : ObjectId("607e9431bc1ab1542d533a65"),
        "kind" : "Traditional vs. contemporary crafts: the distribution of supply\t",
        "kind_en" : "Traditional vs. contemporary crafts: the distribution of supply\t",
        "file" : "Indicatori2",
        "city" : “[cityName]",
        "indicator" : "Culture and craft",
        "values" : [
        {
                "anno" : 2020,
                "type" : "Traditional craft / Folk customs and arts",
                "perc" : 95.7
            }
        ]
    }
```

Craftsmen in tourism

```
source: NoSql
collection: Datas
query: [ All Cities ]
    db.Datas.find({“city”:”[cityName]”, file:”Indicatori3”})
reply:
    {
        "_id" : ObjectId("607e9431bc1ab1542d533a66"),
        "kind" : "Craft activities offering tourism experiences ",
        "kind_en" : "Craft activities offering tourism experiences ",
        "file" : "Indicatori3",
        "city" : "[cityName]",
        "indicator" : "Culture and craft",
        "values" : [
            {
                "anno" : 2020,
                "type" : "Yes",
                "perc" : 4.3
            },

        ]
    }
```

Craftsmen in tourism: timing

Timing by seasonality and timing by frequency

```
source: NoSql
collection: Datas
query: [ All Cities ]
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori4”}) +
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori5”})
reply:
        [indicatori4]
        {
                "_id" : ObjectId("607e9431bc1ab1542d533a67"),
                "kind" : "Craft activities offering tourism experiences by seasonality\t",
                "kind_en" : "Craft activities offering tourism experiences by seasonality\t",
                "file" : "Indicatori4",
                "city" : "[cityName]",
                "indicator" : "Culture and craft",
                "values" : [
                        {
                                "anno" : 2020,
                                "type" : "All year round",
                                "perc" : 100
                        }
                ]
        }

        [Indicatori5]
        {
                "_id" : ObjectId("607e9431bc1ab1542d533a68"),
                "kind" : "Craft activities offering tourism experiences by regularity\t",
                "kind_en" : "Craft activities offering tourism experiences by regularity\t",
                "file" : "Indicatori5",
                "city" : "[cityName]",
                "indicator" : "Culture and craft",
                "values" : [
                        {
```

```
                "anno" : 2020,
                "type" : "Only once or twice a year",
                "perc" : 0
            }
        ]
    }
```

Craftsmen in tourism: experiences offered experience

Offered by a number of experiences and by typology of experiences

```
source: NoSql
collection: Datas
query: [ All Cities ]
    db.Datas.find({“city”:”[cityName]”, file:”Indicatori6”}) +
    db.Datas.find({“city”:”[cityName]”, file:”Indicatori7”})

reply:
    [indicatori6]
    {
        "_id" : ObjectId("607e9431bc1ab1542d533a69"),
        "kind" : "Craft activities offering tourism experiences by number of
            experience offered\t",
        "kind_en" : "Craft activities offering tourism experiences by number of
                experience offered\t",
        "file" : "Indicatori6",
        "city" : "[cityName]",
        "indicator" : "Culture and craft",
        "values" : [
            {
                "anno" : 2020,
                "type" : "Only one tourism experience",
                "perc" : 100
            }
        ]
}

[indicatori7]
{
        "kind" : "Craft actvities offering tourism experiences by kind of experience
                offered (more than one option is possible)\t",
        "kind_en" : "Craft actvities offering tourism experiences by kind of experience
                offered (more than one option is possible)\t",
        "file" : "Indicatori7",
        "city" : "[cityName]",
        "indicator" : "Culture and craft",
        "values" : [
                {
                        "anno" : 2020,
                        "type" : "Guided visits of the headquarters/units or parts of
                                them",
                        "perc" : 0
                }
        ]
}
```

Craftsmen in tourism: main client targets

Main client targets by the number of targets and by typology of targets

```
source: NoSql
collection: Datas
query: [ All Cities ]
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori8”}) +
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori9”})

reply:
        [indicatori8]
        {
```

```
    "_id" : ObjectId("607e9431bc1ab1542d533a6b"),
    "kind" : "Craft activities offering tourism experiences by number of targets
        addressed\t",
    "kind_en" : "Craft activities offering tourism experiences by number of targets
        addressed\t",
    "file" : "Indicatori8",
    "city" : "[cityName]",
    "indicator" : "Culture and craft",
    "values" : [
        {
            "anno" : 2020,
            "type" : "Only one target",
            "perc" : 100
        },

    ]
}

[indicatori9]
{
    "kind" : "Craft activities offering tourism experiences by kind of targets
        addressed (more than one option is possible)",
    "kind_en" : "Craft activities offering tourism experiences by kind of targets
        addressed (more than one option is possible)",
    "file" : "Indicatori9",
    "city" : "[cityName]",
    "indicator" : "Culture and craft",
    "values" : [
        {
            "anno" : 2020,
            "type" : "Families",
            "perc" : 100
        }
    ]
}
```

Craftsmen in tourism: language spoken

Language spoken by a number of languages spoken and by type of languages spoken

```
source: NoSql
collection: Datas
query: [ All Cities ]
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori10”}) +
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori11”})

reply:
        [indicatori10]
        {
                "kind" : "Craft activites offering tourism experiences by number of languages
                        spoken",
                "kind_en" : "Craft activites offering tourism experiences by number of
                        languages spoken",
        "file" : "Indicatori10",
        "city" : “[cityName]",
        "indicator" : "Culture and craft",
        "values" : [
                        {
                                "anno" : 2020,
                                "type" : "Only one languages",
                                "perc" : 100
                        }
                ]
        }


        [indicatori11]
        {
                "kind" : "Craft activites offering tourism experiences by languages spoken
                        (more than one option is possible)\t",
                "kind_en" : "Craft activites offering tourism experiences by languages spoken
                        (more than one option is possible)\t",
                "file" : "Indicatori11",
                "city" : “[cityName]",
                "indicator" : "Culture and craft",
                "values" : [
                        {
                                "anno" : 2020,
                                "type" : "English",
                                "perc" : 73
                        }

                ]
        }
```

Craftsmen in tourism: facilities offered

Facilities offered by a number of facilities provided and by type of facilities provided

```
source: NoSql
collection: Datas
query: [ All Cities ]
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori12”}) +
        db.Datas.find({“city”:”[cityName]”, file:”Indicatori13”})

reply:
        [indicatori12]
        {
                "_id" : ObjectId("607e9431bc1ab1542d533a6f"),
                "kind" : "Craft activites offering tourism experiences by number of facilities
                        offered to visitors",
                "kind_en" : "Craft activites offering tourism experiences by number of
                        facilities offered to visitors",
                "file" : "Indicatori12",
                "city" : "[cityName]",
                "indicator" : "Culture and craft",
                "values" : [
                        {
                            "anno" : 2020,
                            "type" : "Only one facility",
                            "perc" : 40
                        }
                ]
        }
```

```
[indicatori13]
{
        "_id" : ObjectId("607e9431bc1ab1542d533a70"),
        "kind" : "Craft activites offering tourism experiences by kind of facilities
            offered to visitors (more than one option is possible)",
        "kind_en" : "Craft activites offering tourism experiences by kind of facilities
            offered to visitors (more than one option is possible)",
        "file" : "Indicatori13",
        "city" : "[cityName]",
        "indicator" : "Culture and craft",
        "values" : [
            {
                "anno" : 2020,
                "type" : "Dedicated contacts to ask about the activities
                    proposed",
                "perc" : 10
            }
        ]
}
```

The territorial distribution of craft activities: the dynamic map

The distribution and characteristics of craft activities over the urban space, as identified in the previous point

```
source: json files
        a) DynamicMapSets[CityName].json
        b) JansFile[CityName].jspon
```

4.3.4. Environment

How greeny is my city?

Total green areas on total city area: total green area vs. total city area

```
source: NoSql
collection: Datas
query: [ Venezia | Dubrovnik ]
        db.Datas.find({“city”:”[cityName]”, file:”EN1”}) +
        db.Datas.find({“city”:”[cityName]”, file:”EN2”})
reply:
        [EN1]
                {
                         "kind" : "Total green areas",
                        "city" : "Venezia",
                        "indicator" : "Environment",
                        "file" : "EN1", "values" : [
                                [Venezia]
                                {
                                        "anno" : 2010,
                                        "affidamento" : 0,
                                        "adozione" : 2747866,
                                        "qty" : 2747866
                                },
                                [Dubrovnik]
                                {
                                        "anno" : 2010,
                                        "qty" : 159.3
                }
            ]
        }


    [EN2]
        {
            "kind" : "Area totale comune",
            "kind_en" : "Total area",
            "city" : "Venezia",
            "indicator" : "Environment",
            "file" : "EN2",
            "values" : [
                [Venezia]
                {
                    "anno" : 2018,
                    "qty" : 41457.32,
                    "qty_senza_acqua" : 15684.76
                }
                [Dubrovnik]
                {
                    "anno" : 2010,
                    "qty" : 2135
                }
            ]
        }
```

Waste production

Tons of waste production per year/years

```
source: NoSql
collection: Datas
query: [ Ferrara ]
    db.Datas.find({“city”:”[cityName]”, file:”EN2”})
```

```
        [Bari|Venezia|HRCities]
        db.Datas.find({“city”:”[cityName]”, file:”EN3”})
reply:
        [EN2]
        {
                "kind" : "Produzione totale di rifiuti",
                "kind_en" : "Total production of waste per year ",
                "city" : "Ferrara",
                "section" : "Environment",
                "file" : "EN2",
                "values" : [
                        {
                                "anno" : 2005,
                                "qty" : 97120
                        }
                ]
        }
        [EN3]
        {
                "_id" : ObjectId("5ea047425642bf39f9252b05"),
                "kind" : "Produzione totale di rifiuti",
                "kind_en" : "Waste production",
                "city" : "[cityName]",
                "indicator" : "Environment",
                "file" : "EN3",
                "values" : [
                        [Bari | Venezia | Dubrovnik]
                        {
                                "anno" : 2008,
                                "qty" : 199492570
                        }
                        [Sibenik]
                        {
                                "anno" : 2018,
                                "ou" : "Zeleni grad Šibenik d.o.o.",
                "ou_city" : "Šibenik",
                "wc_city" : "Šibenik",
                "type" : "biodegradable waste",
                "qty" : 154
            }
        ]
    }
```

Energy consumption

(kwh / years)

```
source: NoSql
collection: Datas
query:   [Dubrovnik]
         db.Datas.find({“city”:”Dubrovnik”, file:”EN5”})
reply:

         {
                "_id" : ObjectId("5ea040efbd8e9430084d3231"),
                "kind" : "Consumo annuale di energia",
                "kind_en" : "TOTAL ENERGY CONSUMPTION PER YEAR",
                "city" : "Dubrovnik",
                "indicator" : "Environment",
                "file" : "EN5",
                "values" : [
                        {
                                "anno" : 2010,
                                "qty" : 173227155
                        }
                ]
         }
```

Contribution of tourism to waste production

The role of tourism in waste production: per capita waste production on population (kg/person per year) and per capita waste production on population plus tourists

```
source: NoSql
collection: Datas
query:  [AllCities]
        db.Datas.find({“city”:”[cityName]”, file:”EN3b”})
reply:
        {
                "_id" : ObjectId("602a5af5730221305b3a28c1"),
                "kind" : "Ruolo del turismo nella produzione di rifiuti",
                "kind_en" : "Role of tourism in waste production",
                "city" : "[cityName]",
                "indicator" : "Environment",
                "file" : "EN3b",
                "values" : [
                        {
                                "anno" : 2012,
                                "qty" : 565.003738841894,
                                "qty2" : 483.418798797891
                        }
                ]
        }
```

4.3.5. Accessibility and Mobility

Air accessibility and traffic

Evolution of total passengers arriving at the city airport from different origins: total passengers; passengers from international and national flights

```
source: NoSql
collection: Datas
```

```
query:  [Bari                                   |                                   HRCities]
        db.Datas.find({“city”:”[cityName]”, file:”M1”})
reply:
        {
                "_id" : ObjectId("606840d7b81b7f001b5ad2d5"),
                "kind" : "Arrivi in aereo per origine e mese",
                "kind_en" : "Incoming flight passengers by origin and months",
                "city" : “[cityName]",
                "indicator" : "Accessibility and mobility",
                "file" : "M1",
                "values" : [
                        {
                                "anno" : "2019",
                                "mese" : 1,
                                "giorno" : 1,
                                "partenza" : "MILANO LINATE",
                                "arrivo" : "[CityName]",
                                "arrivi_internazionali" : 0,
                                "arrivi_nazionali" : 9,
                                "qty" : 9
                        }
                ]
        }
```

People detected on streets

source: remote API call + json file for polyline

destination: http://137.204.11.71

Mobility Live

Venice and Sibenik run on SQL database, while Ferrara, Bari, and Dubrovnik run on NoSQL database. Please refers to SQL DB details (INSERIRE capitol) and NoSQL DB (CAPITOLO) to better understand the queries structure

The platform used to display the results of the queries is grafana [https://grafana.com], an open-source analytics platform.

source: SQL

database:

Bari: bari

Dubrovnik: DubrovnikPma

Ferrara: Ferrara

Sibenik: sibenik

Venezia: venezia

query:

Bari: to do

Venezia: to do

Note: they probably will be very similar to sibenik solution

Dubrovnik:

Step 1: define networks

```
SELECT DISTINCT `networkId` FROM `Devices`
```

Step2: find stations for networks

```
SELECT * FROM `Devices` WHERE `networkId` = '[network]'
```

Step3: collect from aggregated values in a selected time frame

```
SELECT date_from AS "time",
    SUM(count_unique_asso)
    FROM DevicesStationsCollected
    WHERE date_from > “[value_from]” and date_from < “[value_to]” AND
    id_station IN ([stations_array]) AND
    id_config IN ([interval_config_id])
    GROUP BY date_from
    ORDER BY date_from
```

Ferrara:

Step 1: select stations

```
SELECT * FROM `Stations`
```

Step2: collect from aggregated values in a selected time frame

```
    SELECT
        date_from AS "time",
        count_unique AS "counter"
    FROM DevicesStationsCollected
    WHERE date_from > “[value_from]” and date_from < “[value_to]” AND
    id_station = [station_id] AND
    kind = 'wifi' AND
    id_config IN([interval_config_id])
    ORDER BY date_from
```

Sibenik:

Step1: select barriers from barriers_meta

```
SELECT * FROM `barriers_meta`
```

Step 2: collect from aggregated values in a selected time frame for each barrier: please note: every barrier (station) has 2 “flows”: IN and OUT defined in 2 different barriers (station) with the same name

```
SELECT
    date_from AS "time",
    sum_in AS 'IN',
    sum_out AS 'OUT'
```

```
FROM DevicesStationsCollectedBarriers
WHERE date_from > “[value_from]” and date_from < “[value_to]” AND
AND station_name = [barrier_name]
AND id_config = [interval_config_id]
ORDER BY date_from
```

Step3: select cams from cam_meta

```
SELECT * FROM `cam_meta`
```

Step4: collect from aggragated values in a selected time frame for each cam: please note: every cam has 3 different values in a single time frame:
MIN (the minimum number of people detected in this frame)
AVERAGE (the average number of people detected in this frame)
MAX (the maximum number of people detected in this frame)

```
SELECT
        date_from AS "time",
        sum_min AS 'MIN',
        sum_max AS 'MAX',
        sum_mean AS 'AVG'
FROM DevicesStationsCollectedCam
 WHERE date_from > “[value_from]” and date_from < “[value_to]” AND
AND station_name = [cam_name]
AND id_config = [interval_config_id]
ORDER BY date_from
```

4.3.6. Popularity & Attractiveness

Popularity of the city "brand" in the world

Evolution of searches by “city name” on Google Trends done all over the world: total items vs. “travel“ items

source: NoSql

```
collection: GTrends
query:  [All Cities]
        db.Gtrends.find(
                {
                “city”:”[cityName]”,
                ”area”:”All”,
                ”kind”:”Termine di ricerca”,
                ”categories”:”All”,
                ”term”:”[cityName]”,
                ”year”:{
                        $gte:[anno_from],
                        $lte:[anno_to]
                        }                    })
reply: [all cities ]
        [{
                "_id" : ObjectId("5ef5ac6ce9cfa9600759d54f"),
                "area" : "All",
                "kind" : "Termine di ricerca",
                "city" : "[cityName]",
                "from" : 2004,
                "to" : 2019,
                "categories" : "All",
                "interval" : "month", "year" : [anno_from],
                "month" : 1,
                "value" : 49,
                "term" : "[cityName]"
        }… ]
query:  [All Cities]
        db.Gtrends.find(
                {
                “city”:”[cityName]”,
                ”area”:”All”,
                ”kind”:”Termine di ricerca”,
                ”categories”:”Viaggi”,
                ”term”:”[cityName]”,
                ”year”:{
                        $gte:[anno_from],
                        $lte:[anno_to]
                        }                    })

reply: [all cities ]
        [{
                "_id" : ObjectId("5ef5ac6ce9cfa9600759d54f"),
                "area" : "All",
                "kind" : "Termine di ricerca",
                "city" : "[cityName",
                "from" : 2004,
                "to" : 2019,
                "categories" : "Viaggi",
                "interval" : "month", "year" : [anno_from],
                "month" : 1,
                "value" : 10,
                "term" : "[cityName]"
        }…                                   ]
```

Popularity of the city "brand" in the home country

Evolution of searches by “city name” on Google Trends done in the home country: total items vs. “travel“ items

```
        source: NoSql
        collection: GTrends
        query:  [All Cities]
                db.Gtrends.find(
                    {
                    “city”:”[cityName]”,
                    ”area”:”Italy | Croatia”,
                    ”kind”:”Termine di ricerca”,
                    ”categories”:”All”,
                        ”term”:”[cityName]”,
                        ”year”:{
                                $gte:[anno_from],
                                $lte:[anno_to]
                                }                    })
reply: [all cities ]
        [{
                        "_id" : ObjectId("5ef5ac6ce9cfa9600759d54f"),
                        "area" : "Italy | Croatia",
                        "kind" : "Termine di ricerca",
                        "city" : "[cityName]",
                        "from" : 2004,
                        "to" : 2019,
                        "categories" : "All",
                        "interval" : "month", "year" : [anno_from],
                        "month" : 1,
                        "value" : 49,
                        "term" : "[cityName]"
        }… ]
query:  [All Cities]
        db.Gtrends.find(
                        {
                        “city”:”[cityName]”,
                        "area" : "Italy | Croatia",
                        ”kind”:”Termine di ricerca”,
                        ”categories”:”Viaggi”,
                        ”term”:”[cityName]”,
                        ”year”:{
                                $gte:[anno_from],
                                $lte:[anno_to]
                                }                    })

reply: [all cities ]
        [{
                        "_id" : ObjectId("5ef5ac6ce9cfa9600759d54f"),
                        "area" : "Italy | Croatia",
                        "kind" : "Termine di ricerca",
                        "city" : "[cityName]",
                        "from" : 2004,
                        "to" : 2019,
                        "categories" : "Viaggi",
                        "interval" : "month", "year" : [anno_from],
                        "month" : 1,
                        "value" : 10,
                        "term" : "[cityName]"
        }… ]
```

City attractions for visitors AND Ranking of top cultural attractions for visitors AND “Vibrancy” of the city

Distribution of TripAdvisor city attractions registered in TripAdvisor, by category

```
                source: NoSql
collection:
        Bari: TripAdvisorBari
        Dubrovnik: TripAdvisorDubrovnik
        Ferrara: TripAdvisorFerrara
        Sibenik: TripAdvisorSibenilk
        Venezia: TripAdvisorVenezia

query:  [All Cities]
        db.[collectionName].find({})

reply: [all cities]
        {
                "_id" : ObjectId("606c8b749066ff34a63facba"),
                "id" : "[location_id]",
                "name" : "[location_name]",
                "groups" : {
```

```
    "Shopping" : {
        "Gift And Specialty Shops" : 1
        ...
    }
},
"cats" : {
    "Attraction" : [
        "Shopping"
        ...
    ]
},
"Y2020" : {
    "W20" : {
        "days" : 2,
        "average" : {
            "Reviews" : 0...N,
            "Rating" : 0...5,
            "Photo" : 0..N,
            "Trips" : {
                "business" : 0..N,
                "couples" : 0..N,
                …
            },
            "review_rating_count" : {
                "1" : 0..N,
                "2" : 0..N,
                "3" : 0..N,
                "4" : 0..N,
                "5" : 0..N
            },
            "rank" : 1..N,
            "rankof" : 1..N
        },
        "first_day" : [first day Y-m-d of week 20-2020],
        "last_day" : [last day Y-m-d of week 20-2020]
```

```
                },
                "W21" : {
                        "days" : 7,
                        …
                }
        "Y2021" : { … }
}
```

Lookers vs. bookers in international market

Relationship between the trend of travel-related city searches on Google Trends at world level and the trend of international tourism arrivals in the same period (monthly data)

```
source: NoSql
collection: Datas + GTrends
query:  [Bari | Ferrara]
        db.Datas.find({“city”:”[cityName]”, file:”T11”})
        db.GTrends.find({“area”:”All”, categories:”Viaggi”,city:”[cityName]”,term:”[cityName]”})
reply:
        [T11] + [GTrends]

query:  [Venezia ]
        db.Datas.find({“city”:”Venezia”, file:”A18”})
        db.GTrends.find({“area”:”All”, categories:”Viaggi”,city:”Venezia”,term:”Venezia”})
reply:
        [A18] + [Gtrends]

query:  [HRCities ]
        db.Datas.find({“city”:”[cityName]”, file:”T8”})
        db.GTrends.find({“area”:”All”, categories:”Viaggi”,city:”[cityName]”,term:”[cityName]”})
reply:
        [T8] + [GTrends]
```

Lookers vs. bookers in domestic market

Relationship between the trend of travel-related city searches on Google Trends in the home country and the trend of domestic tourist arrivals in the same period (monthly data)

```
source: NoSql
collection: Datas + GTrends
query:  [Bari | Ferrara]
        db.Datas.find({“city”:”[cityName]”, file:”T13”})
        db.Gtrends.find({
                “area”:”Italy”, categories:”Viaggi”,city:”[cityName]”,term:”[cityName]”
        })
reply:
        [T13] + [GTrends]

query:  [Venezia ]
        db.Datas.find({“city”:”Venezia”, file:”A19bis”})
        db.GTrends.find({“area”:”Italy”, categories:”Viaggi”,city:”Venezia”,term:”Venezia”})
reply:
        [A19bis] + [Gtrends]

query:  [HRCities ]
        db.Datas.find({“city”:”[cityName]”, file:”T10”})
        db.Gtrends.find({
                “area”:”Croatia”, categories:”Viaggi”,city:”[cityName]”,term:”[cityName]”
        )}
reply:
                                        [T10] + [Gtrends]
```

4.4. Direct connect

4.4.1. SQL

Server IP: 94.198.98.184
Port: TCP 3306
The admin provided a username and password to access the server. If not, please request the user's new credentials.
The database is a standard MariaDB server.
To remotely connect the user can use any client that can connect with a MySQL or MariaDB database such as HeidiSQL (https://www.heidisql.com/) or otherwise the user can connect by standard command-line MySQL client.

Each user holds read-only access and is able to view and read just his own DB and Tables.

Bari: db: bari
tables:
- barriers_cnt
- barriers_meta
- CronConfig
- DevicesStationsCollectedBarriers

Dubrovnik: db: DubrovnikPma
tables:
- barriers_cnt
- barriers_meta
- CronConfig
- DevicesStationsCollectedBarriers

Ferrara: db: Ferrara
tables:
- barriers_cnt
- barriers_meta

CronConfig

DevicesStationsCollectedBarriers

Sibenik: db: Sibenik

tables:

barriers_cnt

barriers_meta

CronConfig

DevicesStationsCollectedBarriers

Venezia: db: Venezia

tables:

barriers_cnt

barriers_meta

CronConfig

DevicesStationsCollectedBarriers

4.4.2. NoSQL

Server IP: 94.198.98.184
Port: TCP 27017
The admin provided a username and password to access the server. If not, please request the user a new one.

DBname: symfony

The database is a standard MongoDB server.

To remotely connect the user can use any client that can remotely connect with a MongoDB database such as the official MongoDB tool (https://www.mongodb.com/try/download/compass) or otherwise the user can connect by mongo shell.

Some data collection are “public”, which means every logged user can read:

Data

GTrends

Feeds

Some of the collections are reserved for a specific user:

user: barib-mon

TripAdvisorBari

BariPma

user: dubrovnikd-mon

TripAdvisorDubrovnik

DubrovnikPma

user: ferraraf-mon

TripAdvisorFerara

FerraraPma

FerraraPma2

user: sibeniks-mon

TripAdvisorSibenik

user: veneziav-mon

TripAdvisorVenezia

Please note to log in please remember to declare symfony as your DB. Once logged in, please remember to use symfony as your DB.

Example (from mongo shell client):

```
user@pc:~# mongo -u testuser -p --authenticationDatabase symfony
MongoDB shell version v4.2.10
Enter password: [ password here and press enter ]
…. connecting to: mongodb://94.198.98.184:27017/authSource=symfony&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("58990038-7545-4d01-8fdb-c751e25ce518") }
MongoDB server version: 4.2.10
>
> use symfony [ press enter ]
switched to db symfony
>
```

Now the user can start to query.

4.5. Migrating

It is possible to migrate all the platforms or just a single city.
It is not possible to mirror the platform.

4.5.1. Environment

Server:

O.S.: Debian 9+ or Ubuntu 18

The project has not been tested with other operating systems. However, being based on the symfony4 framework and open source libraries, any environment capable of managing that framework should be able to manage the entire project.

Disk space: at least 20GB for the core.
To properly manage a yearly base data flow, please consider at least 200GB. It strictly depends on the data itself (the amount of data can grow really fast).

RAM: at least 4GB.

Software:

Apache2

mpm_prefork_module
php7_module
proxy_module *
rewrite_module
ssl_module **

*It is recommended to run Apache and Grafana on https on the same server. To do that, it is compulsory to use the proxy module. Since Grafana will share the graphs embedding an iframe, it is compulsory to define a valid URL for grafana on https (i.e. https://grafana.mysqerver.it/grafana) and then define proxy to his standard :3000 port.

Please refer to https://grafana.com/tutorials/run-grafana-behind-a-proxy/

** the project can run without https but it is not recommended.

Database:

The project runs with 2 different databases server, both needed:

- MariaDB (or MySQL DB): there are 6 different DBs defined. To operate the migration, it is necessary at least a database referred to the city chosen.
- MongoDB: to use Mongo DB it is necessary to enable remote connection and authorization. There is just one DB (symfony). To define a collection based ACL, some custom roles have been defined (please refer to this https://docs.mongodb.com/manual/core/collection-level-access-control/)

```
> db.getRoles()
[
        {
                ″role″ : ″roleName″,
                ″db″ : ″symfony″,
                ″isBuiltin″ : false,
                ″roles″ : [ ],
                ″inheritedRoles″ : [ ]
        }
        ...
]

> db.getUsers()
```

```
[
    {
        ″_id″ : ″symfony.barib-mon″,
        ″user″ : ″usernName″,
        ″db″ : ″symfony″,
        ″roles″ : [
            {
                ″role″ : ″roleName″,
                ″db″ : ″symfony″
            }
        ],
        ″mechanisms″ : [
            ″SCRAM-SHA-1″,
            ″SCRAM-SHA-256″
        ]
    }
    …
]
```

Since the admin DB is not public, to obtain read-only access it is compulsory to define roles and users.

Languages:

php 7.3	curl dom gd json mbstring mongodb mysql

	openssl pdo_mysql xml zip the project can run on php 7.4, not tested with php8.
Python 3.5	

Files:

Once the setup phase is complete, it will be possible to copy all the project’s files into the webserver path.
Supposing “/var/www/myProject/” and define the apache virtual host accordingly.

e.g.

```
<VirtualHost *:80>
         ServerName my.url.tld
         DocumentRoot /var/www/myProject/public
         <Directory /var/www/myProject/public>
           Options FollowSymLinks MultiViews
           AllowOverride All
           Order allow,deny
                  Allow from All
         </Directory>
  ErrorLog /var/log/apache2/myProject_error.log
  CustomLog                /var/log/apache2/myProject_access.log                combined
</VirtualHost>
```

Database:

MariaDB:

import the dashboard DB and the city DB.

(choosing to run the platform in a single database, import everything on the same DB).

```
> mysql -u root -p dbNameHere < dashboard.sql
> mysql -u root -p anotherDbNameHere < [city].sql
```

MongoDB:

import the public collections:

- Data
- Feeds
- GTrends

```
> mongoimport -d symfony -c collectionName --file collection.json
```

- and private collections
- TripAdvisor[cityName]

```
> mongoimport -d symfony -c collectionName --file collection.json
```

.env

change at least

```
MYSQL_DATABASE_URL
MONGODB_URL=
MONGODB_DB=
MONGODB_USER=""
MONGODB_PASS=""
```

and the city user and password such as

```
MYSQL_B_USER
MYSQL_B_PASS
MYSQL_B_PASS
```

with the values of the environment

Scheduled tasks:

In order to work properly, set up some cron jobs:

Bari:

```
* */4 * * * user curl http://dashboard.ecipa.eu/bari/getPMA/[apiKey here]
```

Define valid apiKey in BariController.php file (/getPMA @route)

This job calls mongodb and populate mysql tables

```
* * * * *   root    curl http://dashboard.ecipa.eu/bari/cron
```

This job is strictly related to MariaDB's CronConfig table.
It searches a valid interval based on "actual time" and cron rules defined on CronConfig table The table has the same notation of standard cron files to define intervals.

If actual time and config's rule matches, the system aggregate the values with "finestra_secondi" values in which, for example, 9600 means 15 minutes and puts the results into "Collected" table.

Please note that for the City of Bari the option to ask for cam values is not available.
The system will push his records directly into the server calling a web service previously built. To define how to build the web services it is necessary to find an agreement with Bari IT service.

Ferrara:

* * * * * user curl https://dashboard.ecipa.eu/ferrara/getPMA/[apiKey_here]
Define valid apiKey in FerraraController.php file (/getPMA @route).
This job calls a remote web service at 185.31.65.191 and populates MongoDB.
Please refer to 185.31.65.191 web service documentation.

*/10 * * * * user curl http://dashboard.ecipa.eu/ferrara/fromMongo/[apiKey_here]
Define valid apiKey in the FerraraController (/fromMongo @route)
This job transfers some of the MongoDB values into MySQL

* * * * * user curl http://dashboard.ecipa.eu/ferrara/cron
This job is strictly related to MariaDB’s CronConfig table.
It searches a valid interval based on “actual time” and cron rules defined on CronConfig table.
The table has the same notation of standard cron files to define intervals.

If actual time and config’s rule matches, the system aggregate the values with “finestra_secondi” values in which, for example, 9600 means 15 minutes and puts the results into “Collected” table.

Dubrovnik:

* * * * * user curl http://dashboard.ecipa.eu/dubrovnik/getPMA/[apiKey_here]

Define valid apiKey in DubrovnikController.php file (/getPMA @route).
To get a valid API key it is necessary to find an agreement with Dubrovnik's IT service.
This job calls a remote web service at https://api.meraki.com and populates MongoDB and MySQL
Please refer to https://api.meraki.com web service documentation.

* * * * * user curl http://dashboard.ecipa.eu/dubrovnik/cron

This job is strictly related to MariaDB’s CronConfig table.
It searches a valid interval based on “actual time” and cron rules defined on CronConfig table.
The table has the same notation of standard cron files to define intervals.

If actual time and config’s rule matches, the system aggregate the values with “finestra_secondi” values where, for example, 9600 means 15 minutes and puts the results into “Collected” table.

Šibenik:

```
 * * * * *   user    curl http://dashboard.ecipa.eu/sibenik/cron
```

This job is strictly related to MariaDB’s CronConfig table.

It searches a valid interval based on “actual time” and cron rules defined on CronConfig table

The table has the same notation of standard cron files to define intervals.

If actual time and config’s rule matches, the system aggregate the values with “finestra_secondi” values where, for example, 9600 means 15 minutes and puts the results into “Collected” table.

Please note that, since Šibenik directly writes his values into MySQL DB, it is necessary to find an agreement with Šibenik IT service to make them able to write into your DB.

Venice:

```
* * * * *   user    curl http://dashboard.ecipa.eu/venezia/cron
```

This job is strictly related to MariaDB’s CronConfig table.

It searches a valid interval based on “actual time” and cron rules defined on CronConfig table.

The table has the same notation of standard cron files to define intervals.

If actual time and config’s rule matches, the system aggregate the values with “finestra_secondi” values where, for example, 9600 means 15 minutes and puts the results into “Collected” table.

Since Venezia directly writes his values into MySQL DB. You need to find an agreement with Venezia IT service to make them able to write into your DB.