# Smart City vision

an [IASI AI] initiative

# Smart City vision

This document presents the vision of the [IASI AI] community concerning the desired impact and the implementation of Smart City solutions.

## Revision history

23.07.2018 - Initial document release (version 1.0)

## Initiators

Eugen Bușoiu, Gabriel Marchidan, Andrei Pruteanu

## Credits

Photos by Mugur Cardinoiu & Adrian Baltariu
Smart City vision icons by Andrei Modoranu

## Feedback

Send suggestions and feedback over email
**hello@iasi.ai** or **/AI.Iasi**

# Smart City vision

## Business Verticals



## Data



## Historical & real-time data

# Smart City vision

Business Verticals | Vehicle traffic management

## Efficient public transport

Improve transportation system by choosing optimal schedules for each transportation modality (buses and trams)

## Improved daily commuting

Reduce waiting queues by assisting authorities in choosing efficient traffic management strategies during peak hours.

## Integrated traffic management

Real-time updates for stop-light schedules in order to respond readily to dynamic traffic conditions.

# Smart City vision

Business Verticals | Energy consumption management

## Public lighting system

Dynamically start/stop/dim street lights based on external conditions (sunlight, street occupation, crime-rate etc.)

## Buildings (residential/public)

Reduce waiting queues by assisting authorities in choosing efficient traffic management strategies during peak hours.

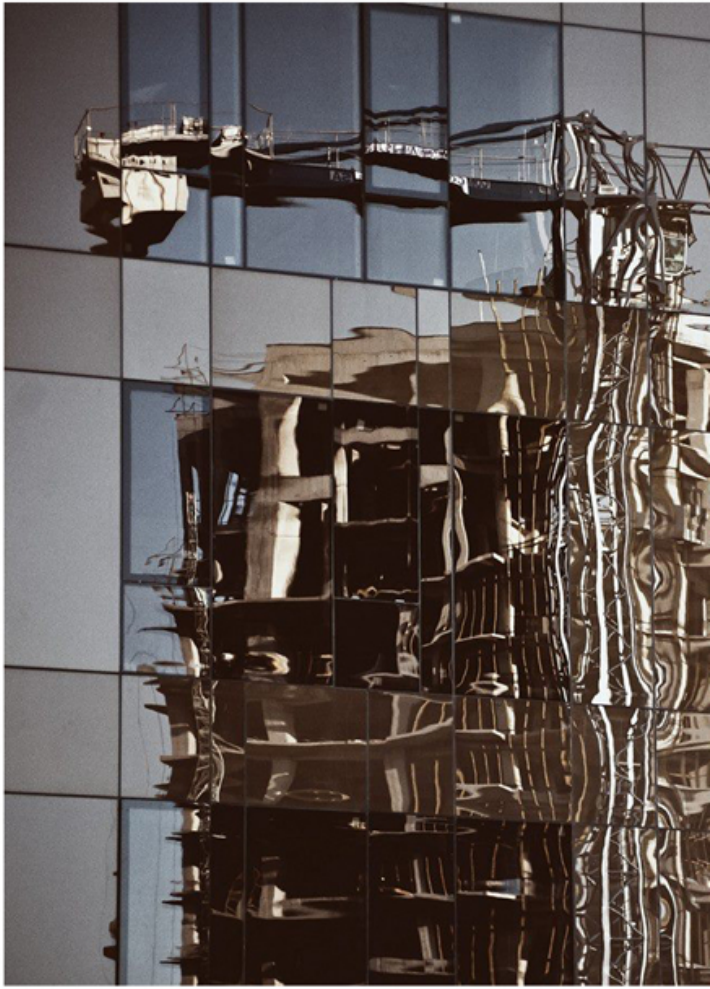
Business Verticals | Urban Planning

## Dwelling (e.g. parks)

Citizens should be able to pick the best schedules for outside activities not only based on weather conditions but also on quality-of-life parameters - air quality, noise level, traffic waiting queues, etc.

## Commuting

By considering realtime and historical trends, city inhabitants can pick up the best schedules for daily commuting.



# Smart City vision

Business Verticals | Urban Planning

## Public space ergonomy (e.g. accessibility)

Public spaces should be organized to best facilitate daily commuting and outdoor activities but also access for persons with disabilities that require special accessibility facilities.

## Construction work

Approved construction activities should be transparent and offered to the public for debating.

Business Verticals | Public Administration

- Expose/store on-going and planned construction-work (e.g. infrastructure repairs/upgrades)
- Up-to-date infrastructure GIS/maps (pipes, cables) - both underground and above ground.
- Transparent budget spending (historical, current and planned)
- Waiting queues at public institutions.
- Make administrative decisions transparent - votes, policy proposals, approved policy documents, decisions, budget, urban planning approvals etc.


# Data

# Smart City vision

## Data Persistency

- All data shall be stored in such a way that single points of failure will be non existent.
- Dives will use RAID with mirroring (RAID 1, RAID 5 etc)
- Databases will be backed up daily.

## Data Availability

- The services will be hosted on infrastructure with high uptime and low latency.
- An uptime greater than 99.9% will be considered acceptable for servers (a total of about 8 hours of downtime per year)
- A latency smaller than 100ms will be considered acceptable for servers (tested from different internet providers in Romania)

## Data Accessibility

- Accessibility to the data shall be provided, upon request.
- Data shall be accessible either by API or by FTP protocol.
- Secure FTP access shall be possible using the SFTP protocol, using generated username and password.
- Data on the FTP shall be accessible exclusively in CSV format. Data on the FTP shall be updated monthly.

# Smart City vision

Data | API

- APIs that expose data shall be exclusively REST APIs.
- APIs shall fully conform to the OpenAPI specification[1].
- API should follow best practices within European Union. CitySDK[2] has already defined ways to expose city information to the public in 6 major cities **Amsterdam** (NL), **Helsinki** (FI), **Istanbul** (TR), **Manchester** (UK), **Province of Rome** (IT), **Lamia** (GR). API should be compliant with CitySDK and follow the same design principles.

Data | Security

- APIs will be accessible only through the **HTTPS** protocol.
- APIs will be accessible only via the use of username/password pairs or API keys.
- FTPs will be accessible only through the **SFTP** protocol.
- FTPs will be accessible through username/password check.
- Authentication shall be provided through **OpenId** connect from a central authentication endpoint.[3]
- Credentials will be unique (not reused) for multiple users. This will enable per-user activity tracking.
- User names and API keys will not be sequential. This will reduce the attack surface.

1. https://github.com/OAI/OpenAPI-Specification
2. http://www.citysdk.eu/
3. http://openid.net/connect/

# Smart City vision

Data | Anonymization

- All the sensible data, where present and applicable, shall be anonymized when made available to the public.
- By sensible data, we define data that uniquely identifies users (such as names, emails, addresses) or data that was not intended to be publicly available (such as passwords, unique citizen identifiers, etc.)
  Access to sensitive information shall follow strict GDPR[4] compliance regulation.

Data | Format guidelines

## CSV

- CSV - Delimited text file used for storing tabular data.
- CSV will be implemented according to **RFC7111**[5] and is the output format that most modern languages implement for CSV.

## JSON

- JSON will be implemented according to **RFC7159**.[6]
- This is the implementation that most modern languages implement for data serialization / deserialization for JSON.
- JSON will only be used as an interchange format and not a persistence format.

4. https://www.eugdpr.org/
5. https://tools.ietf.org/html/rfc7111
6. https://tools.ietf.org/html/rfc7159.html

# Smart City vision

Data | Format guidelines

## JSON-LD

**JSON-LD**[7] (JavaScript Object Notation for Linked Data), is a method of encoding linked data using JSON.

## Geo-JSON

**Geo-JSON**[8] is an open standard format designed for representing simple geographical features, along with their non-spatial attributes. It is based on JSON.

## XML

- XML will be implemented according to **XML 1.0 Fifth edition**.[9]
- This is the implementation that most modern languages implement for data serialization / deserialization for XML.
- XML will only be used as an interchange format and not a persistence format. XML media types shall be configured according to **RFC7303**.[10]

7. https://en.wikipedia.org/wiki/JSON-LD
8. https://en.wikipedia.org/wiki/GeoJSON
9. https://www.w3.org/TR/2008/REC-xml-20081126/
10. https://tools.ietf.org/html/rfc7303

## JSON-LD example:

```
{
   "@context": {
      "smartCity": "http://smartcity.tld/ontology/sc.owl"
   },
   "id": "sm35fba23aaac5dcebd5943d40e2fe9",
   "dataStream": []
}
```

## Geo-JSON example:

```
{
   "type": "FeatureCollection",
   "features": [
      {
         "type": "Feature",
         "properties": {},
         "geometry": {
            "type": "Point",
            "coordinates": [
               27.586798667907715,
               47.15768061059638
            ]
         }
      }
   ]
}
```

## Relational databases

**All relational databases shall observe the guidelines presented below.**

### Primary Key

- Can not be a multipart field.
- Must contain unique values.
- Cannot contain null values.
- Its value in not optional in whole or in part.
- It comprises a minimum number of fields necessary to define its uniqueness.
- Its values must uniquely and exclusively identify each record in the table.
- Its value must exclusively identify the value of each record within a given record.
- Its value can be modified only in rare or extreme cases.
- Each table must have one, and only one, Primary key.
- Each Primary Key within the database must be unique, no two tables should have the same primary key unless one of them is a subset table.

### Foreign Key

- Has the same name as the Primary Key from which it was copied.
- Uses a replica of the field specifications for the primary key from which it was copied.
- Draws its values from the primary key to which it refers.

## Relational databases

### Table Field

- It represents a distinct characteristic of the subject of the table.
- It contains only a single value.
- It cannot be deconstructed into smaller components.
- It does not contain a calculated or concatenated value.
- It is unique within the entire database structure.
- It retains the majority of its characteristics when it appears in more than one table.

### Database Field

- It represents a single subject, which can be an object or event.
- It has a Primary Key.
- It does not contain multipart or multivalued fields.
- It does not contain calculated fields.
- It does not contain unnecessary duplicate fields.
- It contains only an absolute minimum amount of redundant data.

Suggested RDMS[11] (Relational Database Management Systems), in no particular order: MySql, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server. Relational databases shall be used only as a persistence format.

11. https://en.wikipedia.org/wiki/Relational_database_management_system

# Smart City vision

## Data Collection guidelines

- Data shall be transformed from any initial format to either csv or relational database.
- Extra metadata shall be added, where available - most notably the location where it was measured and unique equipment id, where available.

## Data Aggregation guidelines

- Primarily, data will be stored in its raw format.
- From this format, more specific views or subsets will be extracted is so-called working sets.
- Working sets will be anonymized (if needed) and will contain only the fields relevant to the intended use of the dataset. From each data source one or more working sets will be generated.
- Working set extraction will be automated for each data source. This will enable good reactivity if any of the working sets will need to be regenerated (either because of new data being added or a corruption of the existing working sets).

# Historical & real-time data

# Smart City vision

Historical & real-time data | Historical

- All data should be stored and offered for download via the public API and/or **SFTP**.
- Information retrieval should support time-slicing.
- Users should be able to download desired historical information from date X to date Y.

Historical & real-time data | Real-time sensing

These are examples of important data to be provided via the public API.

| Data Signal | Meaning | Data type | Physical Measure | Sensor Mobility |
|---|---|---|---|---|
| Energy Consumption | Electrical | float | MWh | fixed |
| | Gas | float | MWh | fixed |
| Water Consumption | Pressure | float | Bars | fixed |
| | Chemical compounds | float | Nitrates | fixed |
| Air Quality | PM 2.5, PM 10 | int | Micro-particle level | fixed / mobile |
| | $NO_2$, NOx, $SO_2$, benzen, CO, $CO_2$, VOC | int | Grams | fixed / mobile |
| Noise Level | Sound level | int | dB | fixed / mobile |

# Smart City vision

Historical & real-time data | Real-time sensing

| Data Signal | Meaning | Data type | Physical Measure | Sensor Mobility |
|---|---|---|---|---|
| Waste Collection | Collected | int | Tons | fixed |
| | Processed | int | Tons | fixed |
| Vehicle Traffic | Counter | int | N/A | fixed |
| Public Transport | Arrival delay | int | Seconds | fixed |
| Parking Spots | Occupancy | int | Availability | fixed |
| Flash Crowds (during public events) | Counter | int | People count | fixed |
| Public Spaces Availability (congress rooms, free beds in hospitals) | Occupancy | int | Availability | fixed |

# Smart City vision

Value-added services (via AI) | Data analytics (City Dashboard)

## Historical Trends & Analysis

- Measured at various time intervals (hourly, daily, weekly, seasonally, yearly)
- How air quality changes based on weather conditions, seasons and vehicle traffic.
- How does water quality/pressure change based on seasons, weather and various pollution incidents.
- In what way aggregate waste collection/recycling changes based on weekly cycles, seasons and/or local policies (prices / taxes)
- How does the noise levels change through the city based on vehicle traffic and hour of the day.
- In what way energy consumption evolves through the city based on hour of the day, outside weather and seasons.
- How does public transport waiting time changes based on hour of the day and weekday.
- How does criminality rate change through the city and with what it correlates to.
- Impact assessment for city management actions (e.g. how air quality has improved/degraded given traffic management actions - allow/disallow heavy vehicles powered by diesel engines in certain areas such as historical city center)

# Smart City vision

Value-added services (via AI) | Data analytics (City Dashboard)

## Forecasting

- Predict air quality given weather forecast and historical data.
- Predict commute time and traffic queues given historical patterns and specific street conditions.
- Predict public energy consumption given weather conditions and historical patterns.
- Predict human mobility patterns (daily, seasonally) for various urban setups.
- Predict waste collection improvement given administration rules and historical data.
- Predict water pressure at source based on seasonal trends and influence of weather conditions.
- Improve scheduling for commuting, dwelling and public transport. Artificial Intelligence offers advanced optimization algorithms to improve overall traffic scheduling.
- Correlate power failures with with other measurements such as weather conditions and industrial activities (e.g. ongoing construction-works)

## Event Notifications

- Sensor readings above certain thresholds (air pollution, vehicle traffic, peak energy/water consumption, noise etc.)
- Ahead-of-time failure notifications (e.g. pipes, cables, street lights, etc.)
- Face recognition for security considerations (e.g. detect criminal activities)
- Real-time hazardous situation detection during public events (e.g. flash-crowd detection via WiFi fingerprinting)

# Smart City vision

Value-added services (via AI) | Data analytics (City Dashboard)

Once the data is available programmatically, there a large number of possible services that can be built on top.
Here are some examples:

- Resource availability and registration (e.g. public parking spots, conference rooms, municipal counselor availability, public events, free beds in hospital, etc.)
- A real-time map of ongoing construction-work (e.g. closed streets or restricted vehicle traffic, infrastructure downtime/maintenance - water/gas/electricity) and ahead-of-time planning for such maintenance activities.
- Suggest traffic re-routing to avoid polluted areas.
- Neighborhood evaluation for urban planning (public/private construction work, leisure activities, etc.)
- Instant feedback to authorities via mobile applications (complaints, suggestions, alerts, etc.)
- Public participation in budget planning - citizen feedback.
- Public polling - ask citizens for various development scenarios.
- Inform citizens about policy changes (e.g. changing taxes, public events, etc.)

[IASI AI]