Auditing and Assessment of data traffic flows in an IoT Architecture

Authors: P. Nesi, G. Pantaleo, M. Paolucci, I. Zaza

Speaker
Michela Paolucci

Department of Information Engineering, University of Florence
Via S. Marta 3, 50139, Firenze, Italy

http://www.disit.dinfo.unifi.it
michela.paolucci@unifi.it
Disit Roadmap

2013
- Km4City 1.1
- Tuscany Map
- Services
- AV
- Sensors
- Parking

- Cultural Heritage
- Enrichment cities
- Event in the city
- Digital Locations
- Fresh places

2014
- Weather
- Cultural Heritage
- Energy recharge pillar

- Wi-Fi
- Events in the city

2015
- Km4City 1.4
- Embed
- More API
- iBeacon

2016
- Sii-Mobility
- SCN
- 2016-2018 - Started

2017-2020
- RESOLUTE
- 2015-2018 - Started

2018
- GHOST SIR
- 2016-2019 - Started

- REPLICA
- 2016-2021 - Started

- REPLICA
- Monitoring
- Smart City

- SELECT
- Monitoring
- Smart City IOT

- 5G
- Smart City vs IOT, Industria 4.0

Already Planned up to 2021

- Waste
- Territorial areas and paths
- Health, Bike sharing
- Statistics, Energy, ICT, ...
- E-vehicles
- Risk analysis
- Environmental, water
- Data Licensing models
- Energy Meters
- Fi-Ware compliant

- More Sensors, IOE, IOT
- Dashboard Builder
- Territorial areas and paths
- User Engagement
- Mobility and transport
- Resilience Decision Support

- Suggestions on demand
- User Behaviour Analysis
- Trajectories and OD

The 4th IEEE International Conference on Collaboration and Internet Computing
Smart Cities / IoT Solutions features

• Collect Big Data from the cloud of city sensors and IoT devices

• Provide to final users a set of tools and items to easily access, read and monitor the ingested data

• Give the possibility to build applications to visualize, process and perform different kinds of data analytics
Smart City Requirements (1)

• I) Necessity of minimizing the problems induced to data-driven applications:
  – storing continuously last values from all devices
  – collecting historical trends generated by the ingestion of last values cumulated over time (data caching on IoT)
  – enriching ingested data (LOD, SC semantic KBs, ontologies and repositories)

• II) Necessity of quantitatively monitoring messages/data flows in order to detect anomalies and problems in data traffic
Data Flows (1)

- **IoT Devices**
  - providing/receiving data to/from: IoT Brokers, IoT Apps, control messages to IoT devices (e.g. red lights, video cameras, signage, etc.) mainly via http carrying other protocols such as MQTT, NGSI, etc.

- **IoT Brokers**
  - providing/receiving data in push/pull

- **Data Transformation Processes**
  - ETL (Extract, Transform and Load) processes for data warehousing, which collect data in pull/get or through http/https/ftp/ftps/WS-soap, etc., protocols
Data Flows (2)

• Data stream processing data coming from
  – video cameras and other several kinds of sensors
    (weather, air quality, traffic flows, levels of water in rivers
    and underpasses, etc.)

• Dashboards
  – web representation of data regarding the city status that
    can be represented in several control rooms

• Resource and Data portals
  – provide, receive and manage a set of resources

• Database storage
  – MySQL, NoSQL, HBase, Mongo, Virtuoso, etc.
Snap4City Architecture

The general purposes of the Snap4City system are:

- Collect data from external sources
- Semantically aggregate
- Exploiting data by data analytics and produce new knowledge and services for the users:
  - Traffic flow reconstruction
  - Routing
  - Predictions: on free parking, on traffic flows
  - Points of Interest (POI)
  - Smart City services: bike sharing, first aid, public transportation, smart waste management, air quality monitoring, weather reports and predictions, city events and entertainments, etc.
Snap4City Architecture (2)

• Data to be collected come from: OD, RT data, personal data, IoT/IoE, stream, data driven, Industry 4.0, Social Media, etc. (any kind of source, protocol and format)

• Consists of a set of tools to cope with the whole process of ingestion, collection, analysis of data
Snap4City tools for monitoring traffic flows

• The Snap4City system deals with large amounts of IoT devices generating a large number of communication and data flows

• It is necessary to monitor the proper working of the whole system and provide tools to understand potential malfunctions and recover the fully working conditions

• To this end, dedicated tools have been designed and implemented in the Snap4city platform:
  – DevDash and AMMA
Developer Dashboard (DevDash) tool

• Developer Dashboard (DevDash) tool is a data Value control tool aiming at:
  – collecting, enriching and indexing data coming from IoT devices

• Application & Microservices Monitoring and Analyzer services (AMMA) tool
  – is a data flow control tool for real-time monitoring and analyzing traffic flows

• DevDash and AMMA share the same back-end framework
DevDash and AMMA back-end Architecture

- Input data are collected in real-time by:
  - Persistent storage (NoSQL database)
  - IoT Brokers (which store the current value of IoT device until it is consumed)
  - Directly from the IoT devices.
DevDash and AMMA back-end Architecture

- The EventLogger role is to study the resource consumption of services and applications.
- Each Snap4City application /service, finalized to data collection, is provided with a dedicated logging block sending event-driven or scheduled messages to the EventLogger (APIs exploiting the RSyslog protocol).
- All data the ingested are processed by a module exploiting the open source Apache NiFi tool.
DevDash and AMMA back-end Architecture

- Apache NiFi is used to manage and optimize Big Data streams, operating also data transformation to properly prepare and adapt data for further processing, storage and indexing.
- The obtained data stream is dispatched to Data-Shadowing, the Traffic Data and the Data Enrichment modules for: additional processing, storing (Apache HBase) and indexing.
- The frontends are based on Banana Web-app, an open source data visualization tool developed as a port of Kibana.
DD & AMMA – Visual Tools (1)

- **DevDash:**
  - Allows users to apply filters in succession, until they reach the data view of interest for their purposes: i) download data; ii) consult data details; iii) send data to Data Analytic tools

- **AMMA**
  - Monitors and controls all the communication activities in Snap4City, in terms of data flows and traffic
  - Allows to make analysis of the indexed EventLogger data for exploring distinct aspects of the data flows
DD &AMMA – Visual Tools (2)

• AMMA and DevDash includes a set of dashboard widgets:
  – histograms, time-pickers, filters, facet selection on the different kind of data managed, heatmaps, pie charts, tables, and also newly created panels, such as a SC linked map with geo-faceting graphical filtering capabilities

• All these widgets are useful to perform real-time drill down activities on the data flows:
  – panning/zooming in time, in space, and navigating in kind via faceted solution and indexes, searching in text in a certain context, in terms of relationships among entities, etc. to arrive at the end at precise selection, at the single value on the graph or table

• Solr facet search functionalities provides the real-time analytic processing layer which is required for producing dynamic visualizations and different views on data
DevDash Case Study (1)

• Check the status of IoT devices and related data traffic

• Drill-down on single data related to a single IOT:
  – filtering on time: the single data portion corresponding to the unexpected data flow can be viewed

Requirement I solved:
Accessing the last value and real time values measured by IoT devices
DevDash Case Study (2)

- Detect potential anomalies or disfunctions by inspecting the DevDash tool time trend.
AMMA Case Study (1)

- Monitor data traffic flows among IoT devices, services, applications etc. and detect potential anomalies

- Unexpected behaviours can be revealed by inspecting the data flow time trend:
  - a) detecting peaks or valleys in the trend
  - b) drill-down on data to identify single/more malfunctioning devices and/or services

Requirement II solved: quantitatively monitoring data/message traffic and flows
AMMA Case Study (2)

• Make drill-down activities on data related to a single Process ID and check for unexpected behavior in the Time Trend panel:
  – c) Filtering data by the Process ID (e.g. for example those related to a SmartWaste container)
  – d) Detect a peak with more data traffic than expected during its scheduled activity, by properly filtering on time, the single data portion corresponding to the unexpected data flow can be viewed
  – e) Locate on map the single involved device or service
Auditing and Assessment of data traffic flows in an IoT Architecture

Authors: P. Nesi, G. Pantaleo, M. Paolucci, I. Zaza

Speaker
Michela Paolucci
Department of Information Engineering, University of Florence
Via S. Marta 3, 50139, Firenze, Italy
http://www.disit.dinfo.unifi.it
michela.paolucci@unifi.it