Emerging Distributed Computing and Challenges for Services Engineering

Hong-Linh Truong
Distributed Systems Group, TU Wien

truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong
@linhsolar

Goals

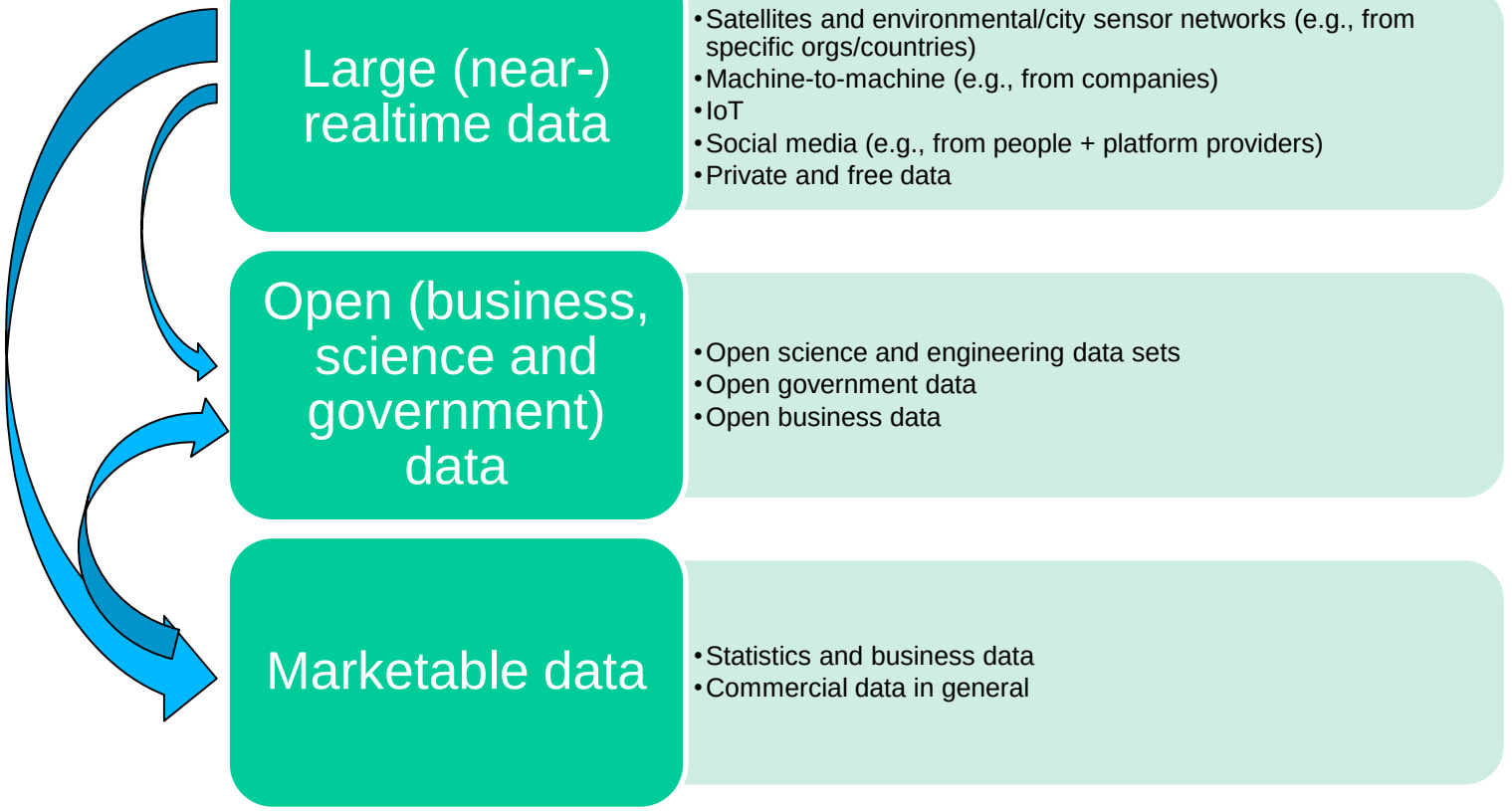
- See emerging trends in distributed systems and computing
- Have a critical look at use cases and analyze use cases
- See the service engineering technologies needed for such use cases

Outline

- Some emerging models – properties and issues
  - IoT resources
  - Data provisioning models
  - Computational infrastructures/frameworks provisioning
  - Human computation provisioning
  - Software-defined *
- Use cases
- Advanced services engineering
  - Single service/platform engineering
  - Multi-platform services engineering

WHICH ARE EMERGING FORMS OF DISTRIBUTED COMPUTING MODELS, SYSTEMS AND APPLICATIONS THAT YOU SEE?

Emerging data provisioning models

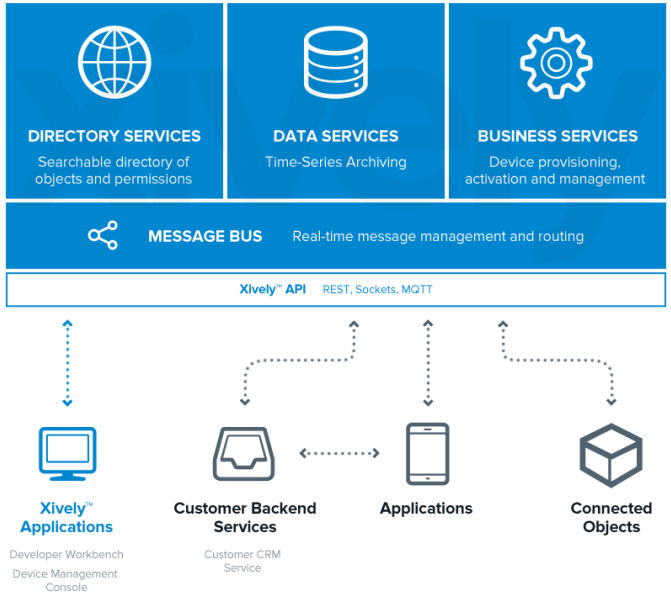


Data are assets

Examples of large-scale (near-) realtime data

PACIFIC CONTROLS
Galaxy
PLATFORM OF PLATFORMS

Xively Cloud Services™
https://xively.com/



OOI OCEAN OBSERVATORIES INITIATIVE
CYBERINFRASTRUCTURE
Providing a link between ocean research and discovery

Large-scale (near-)realtime data: properties and issues

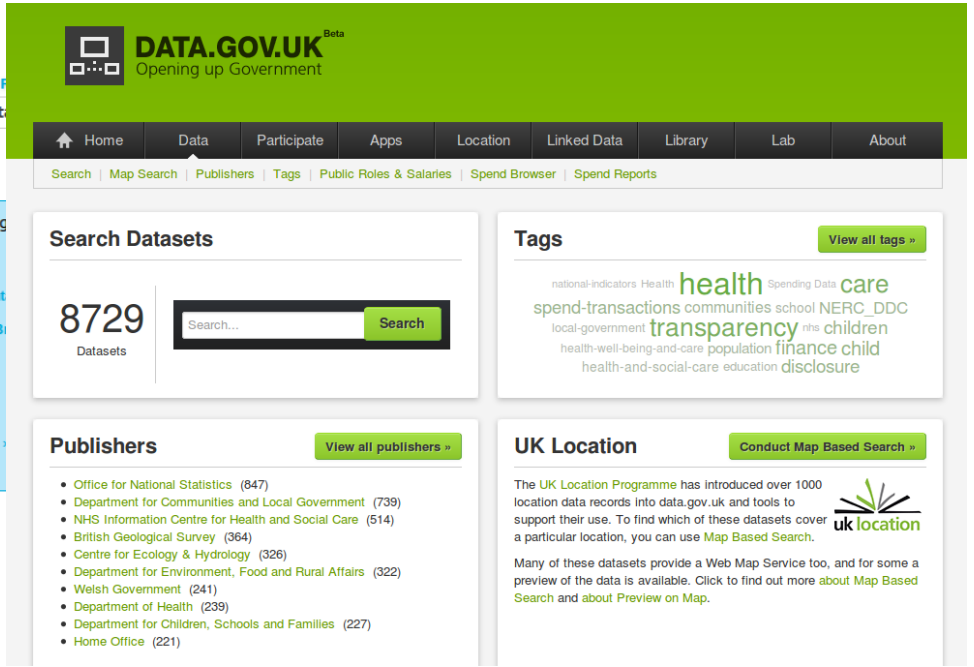
Some properties

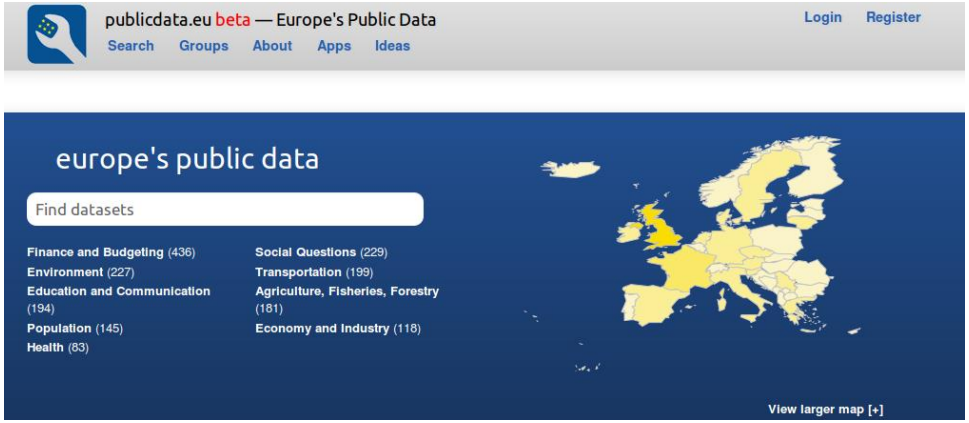
- Having massive data
- Requiring large-scale, big (near-) real time processing and storing capabilities
- Enabling predictive and realtime data analytics

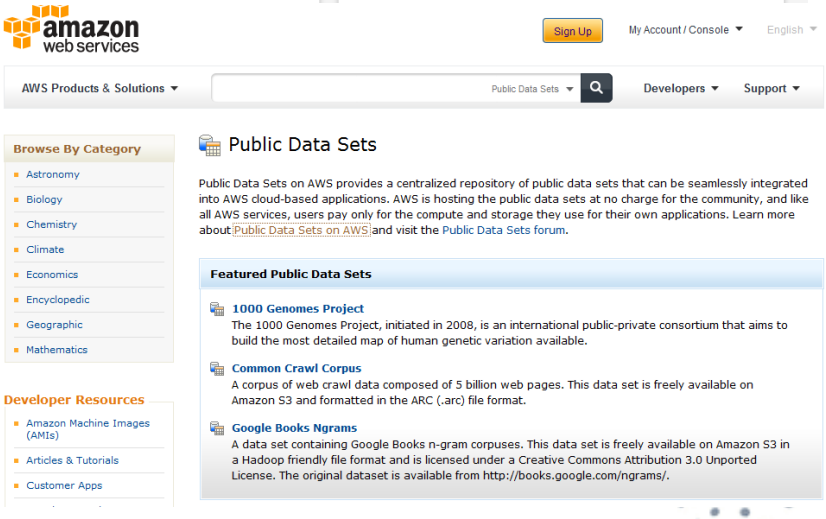
Some issues

- Timely analytics
  - Performance and scalability
- Quality of data control
- Handle of unknown data patterns
- Benefit/cost versus quality tradeoffs

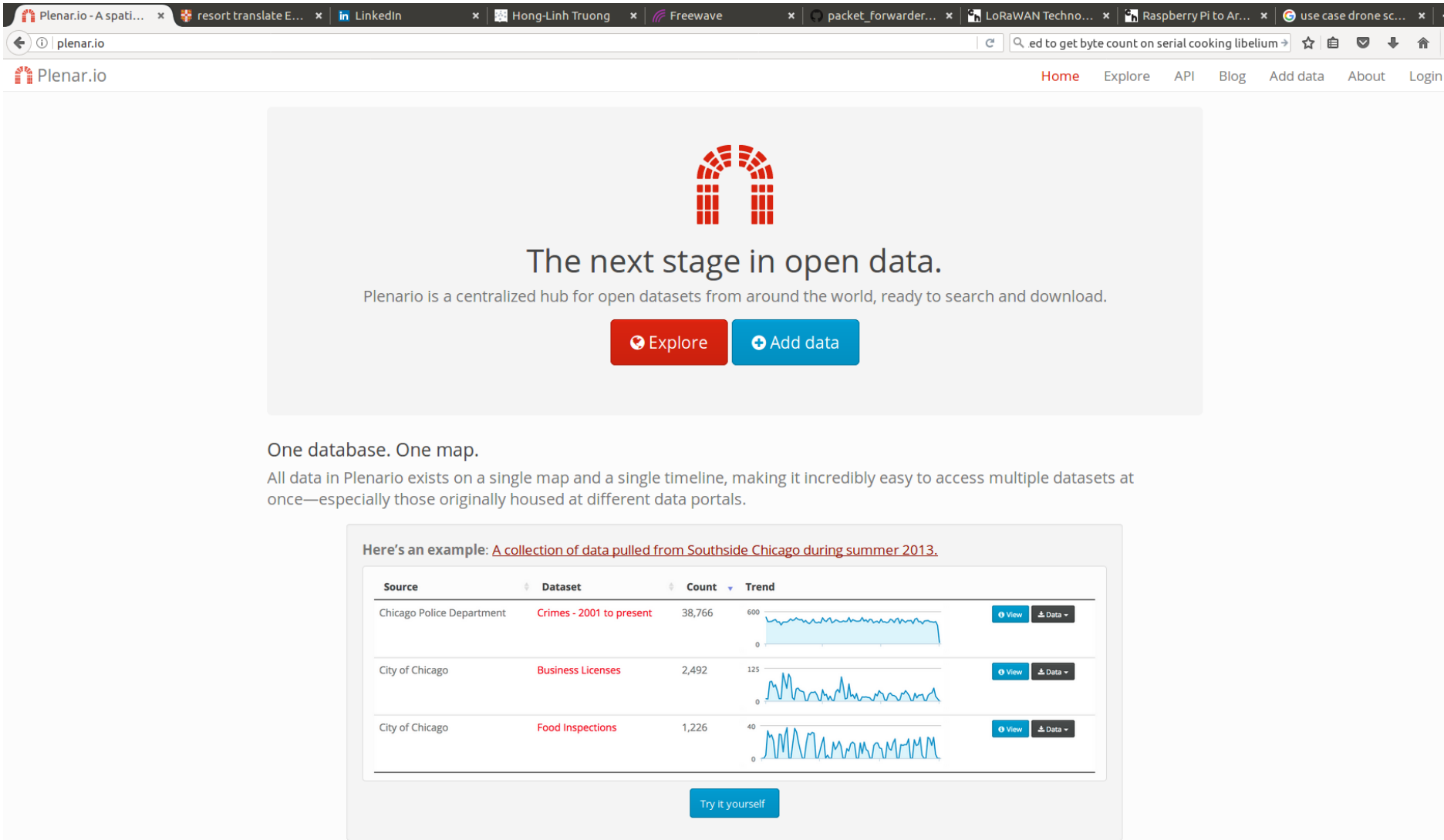
Example of open data





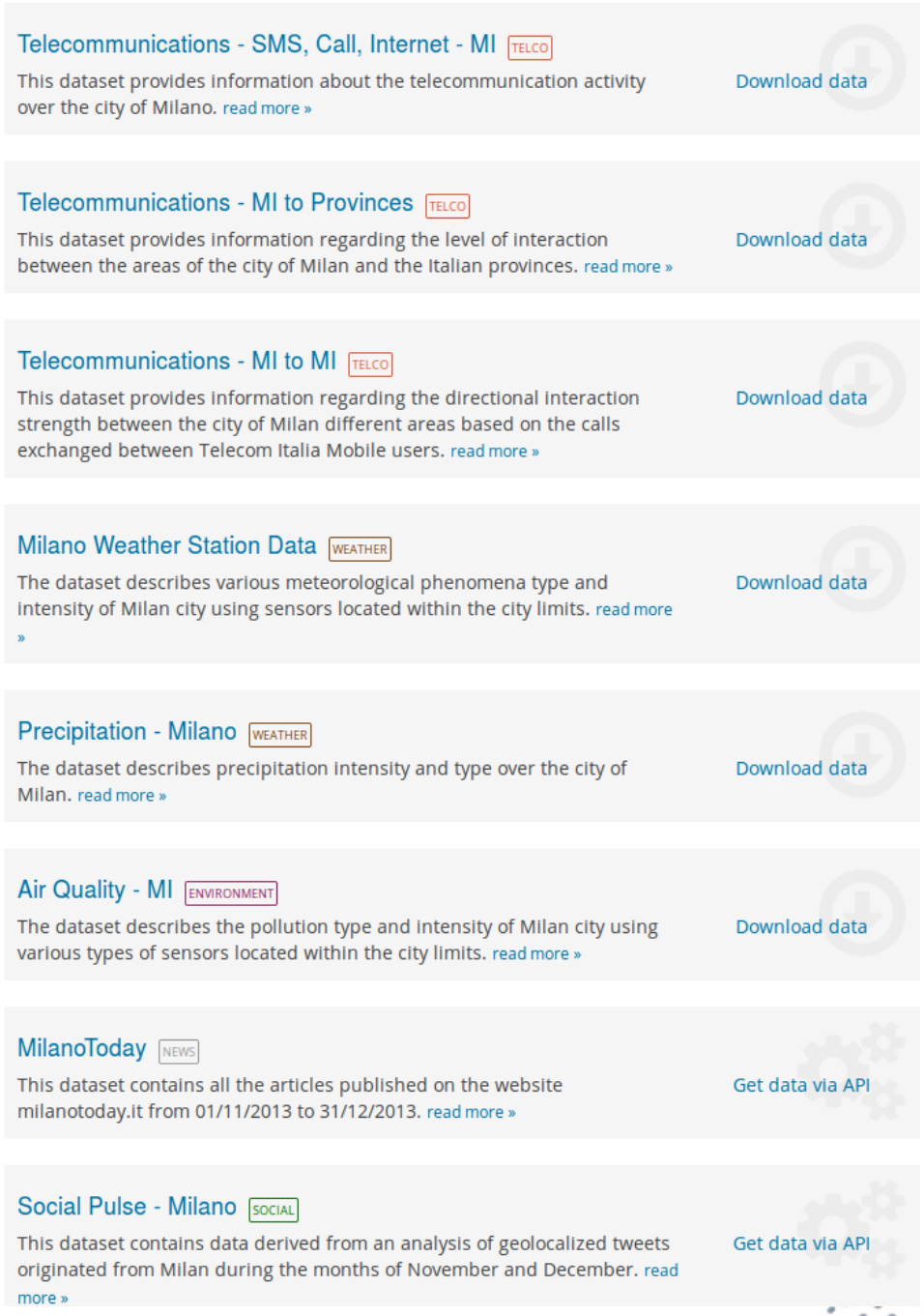


Static + Realtime Open Data



Telecommunication

https://dandelion.eu/datamine/open-big-data/



Open data: properties and issues

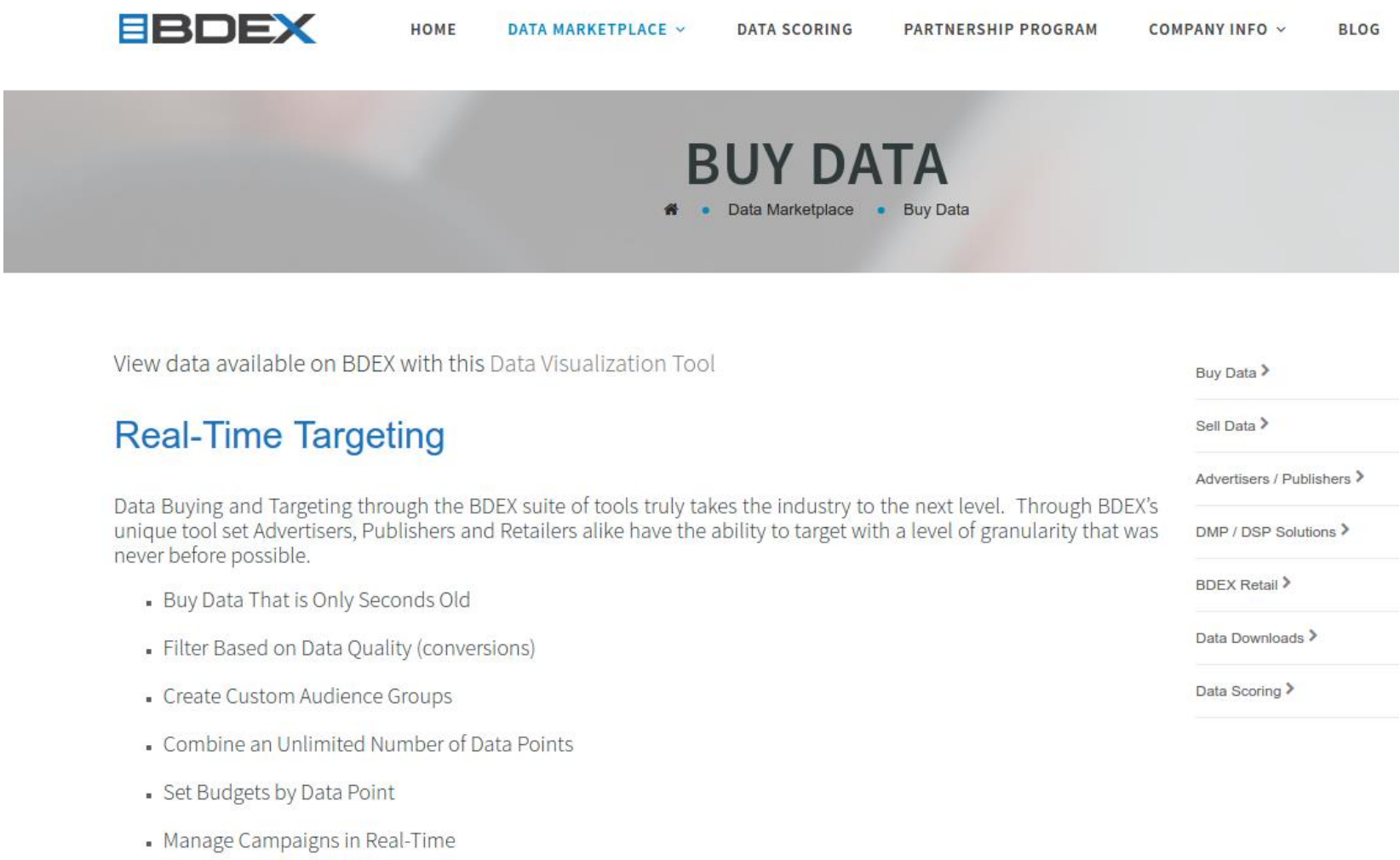
Some properties

- Having large, multiple data sources but mainly static data
  - Real-time, open data is growing
- Having good quality control in many cases
- Usually providing the data as a whole set

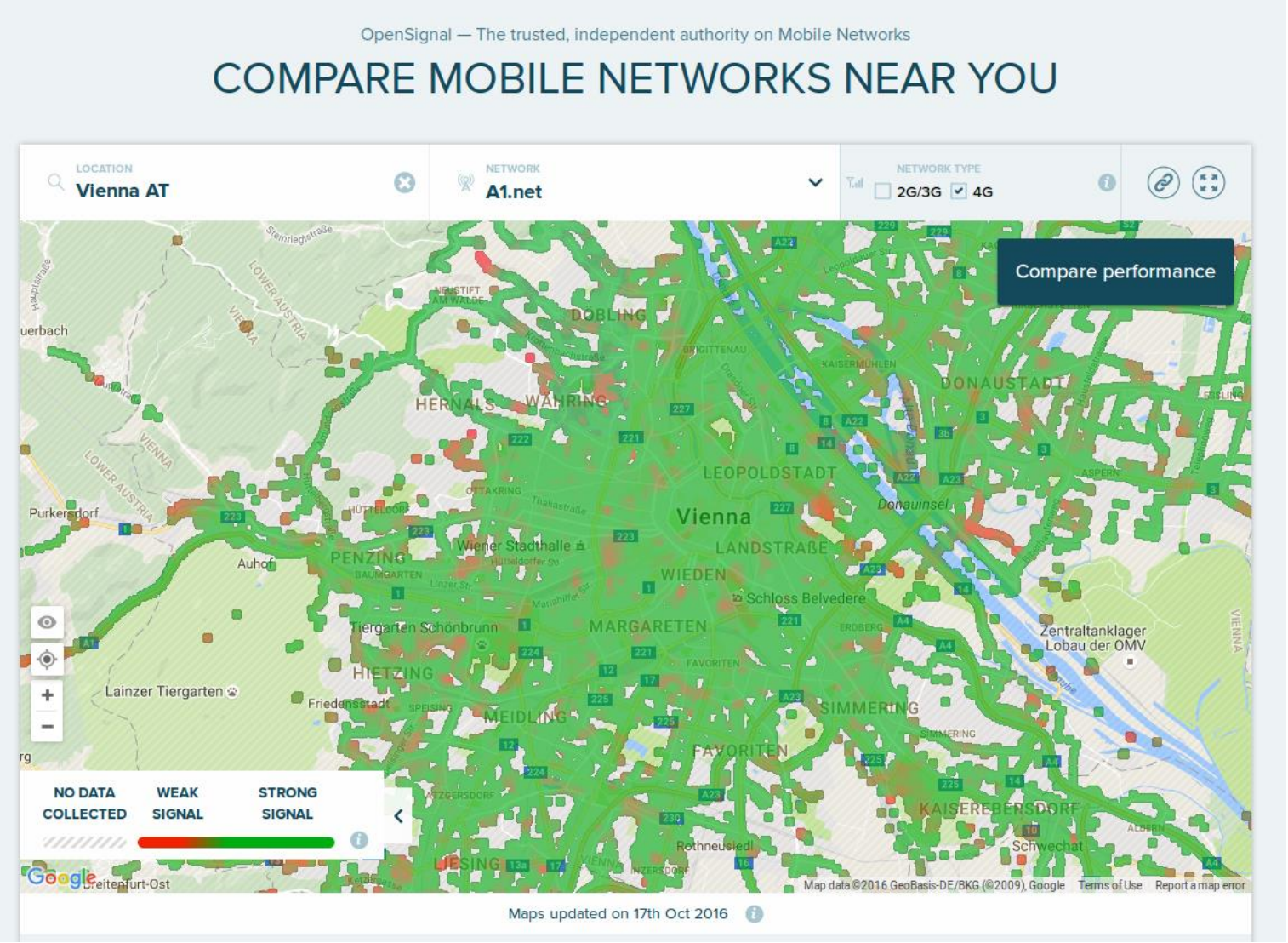
Some issues

- Fine-grained content search
- Balance between processing cost and performance
- Correlation/combination with real-time/private data

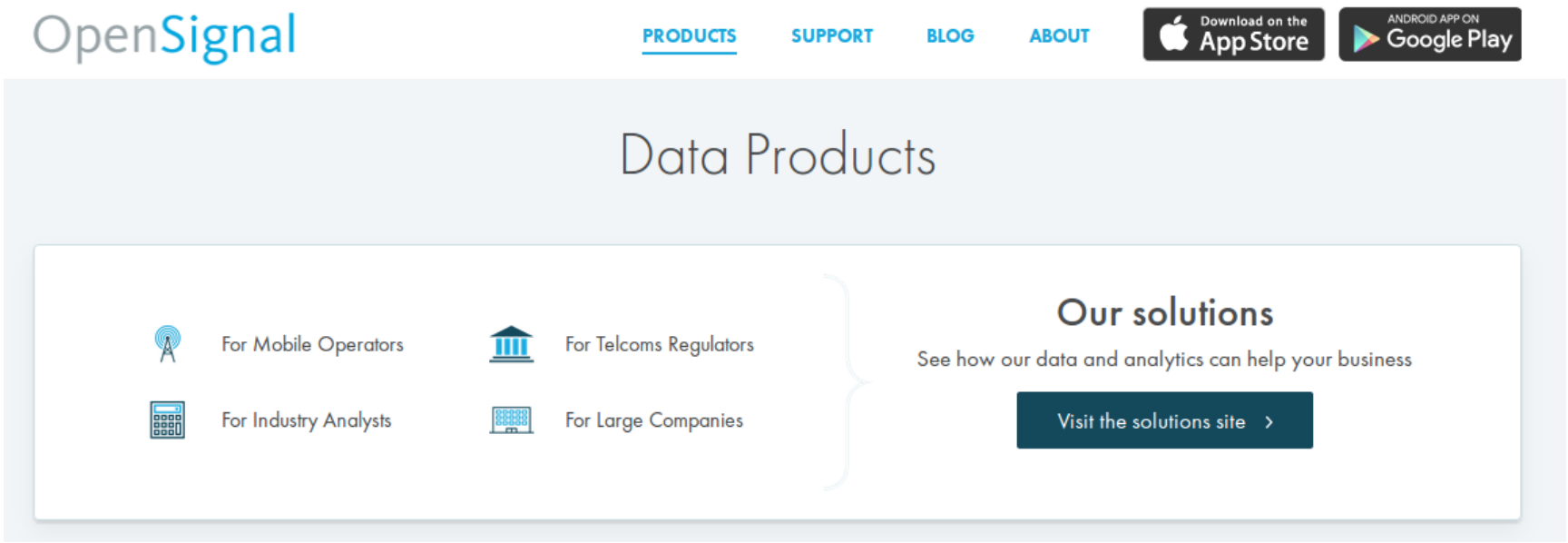
Marketable data examples



Marketable data examples



Marketable data examples



Marketable data: properties and issues

Some properties

- Can be large, multiple data sources but mainly static data
- Having good quality control
- Have strong data contract terms
- Some do not offer the whole dataset

Some issues

- Multiple levels of service/data contracts
- Compatible with other data sources w.r.t. contract
- Cost w.r.t. up-to-date data
- Near-realtime data marketplaces

Emerging computing infrastructure and platform provisioning models

- Infrastructure-as-a-Service
  - Machine as a service
  - Storage as a Service
  - Database as a Service
  - Network as a Service (think about Network Function Virtualization with 5G)

Emerging computing infrastructure and platform provisioning models

- Platform-as-a-Service
  - Application middleware
  - Computational frameworks
  - Data processing frameworks
  - Management middleware (e.g., monitoring, control, deployment)
- Technologies
  - Virtualization
  - Microservice architectures
  - Serverless computing
  - Etc.

Examples

Data Processing Framework

- Amazon Elastic MapReduce
- Cloudera Impala
- Apache Kylin
- Google Cloud Dataflow
- Summingbird
- Azure Stream Analytics

Data Transfer/Messaging Middleware

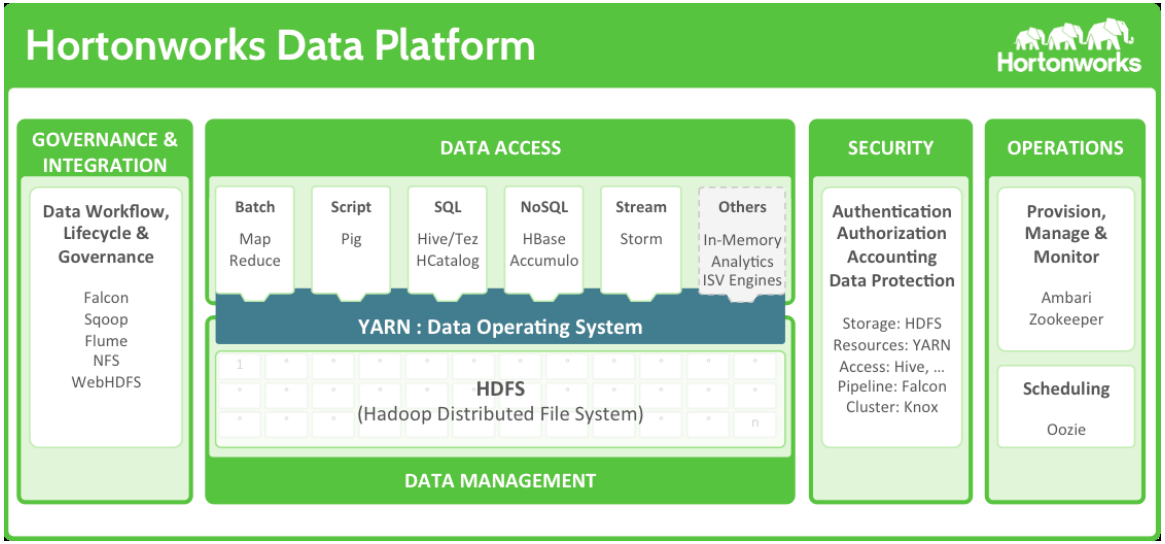
- StormMQ
- Apache Nifi
- Kafka
- MQTT
- AmazonSQS
- CloudAMQP

Data Storages

- Google BigQuery
- Elastic Search
- Influx DB
- MongoLab
- Amazon S3
- Cassandra

Hadoop ecosystem

- Built around Mapreduce programming models and Hadoop software ecosystems
  - http://hadoop.apache.org/
- From “The Forrester Wave™: Big Data Hadoop Distributions, Q1 2016”: Top Hadoop solution providers are Cloudera, Hortonworks, IBM, MapR Technologies, and Pivotal Software



Source: http://hortonworks.com/blog/defining-enterprise-hadoop/

Spark ecosystem

Programming with Java, Scala, Python, R

We can have a separate modules

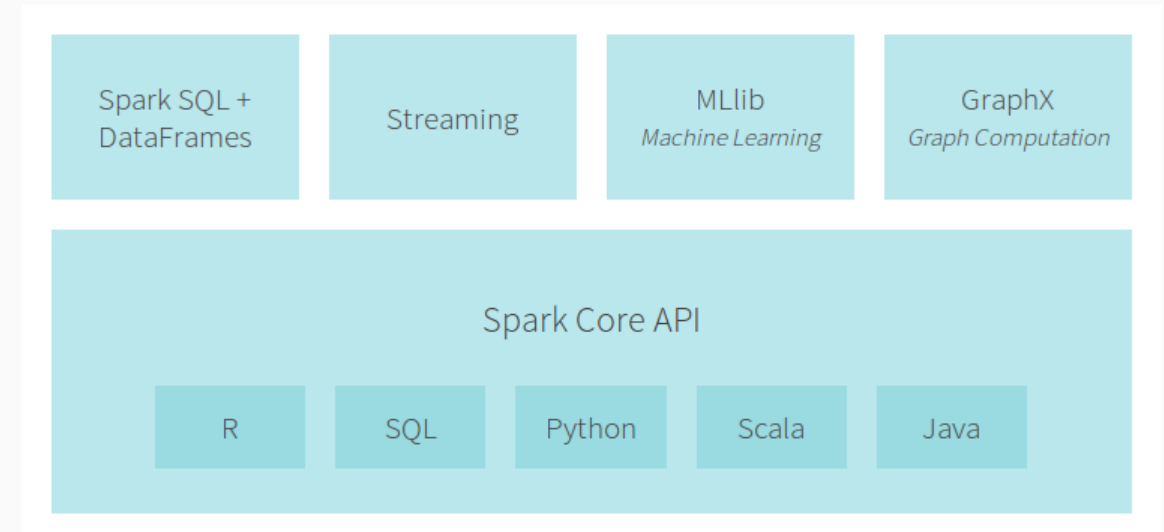
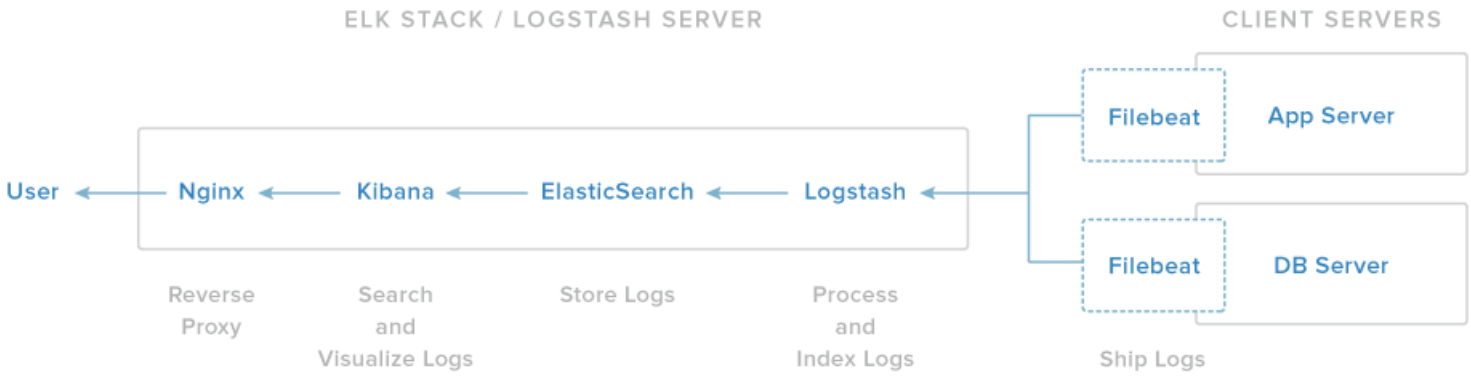


Figure source: https://databricks.com/spark/about

ELK Stack

Building using elastic components: Elasticsearch, Elasticsearch Hadoop, Kibana, and Logstash

https://www.elastic.co/



Source: https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elk-stack-on-ubuntu-14-04

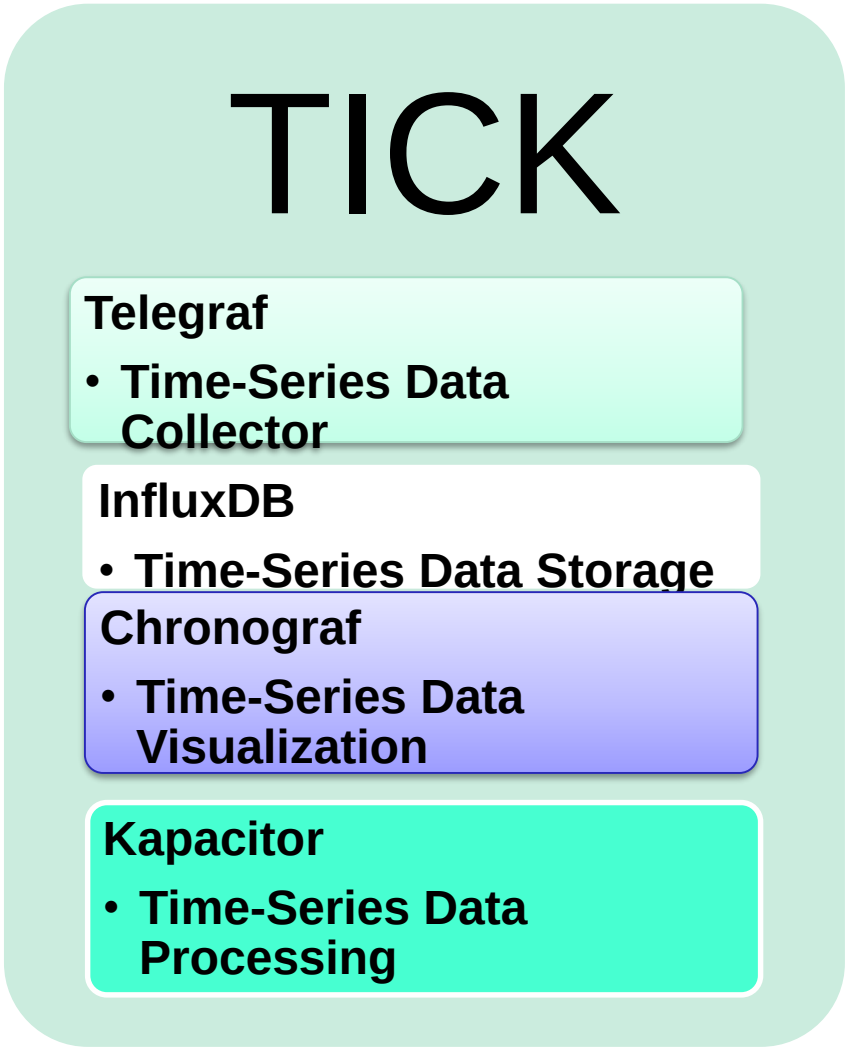
TICK Stack

Main from services of Influx

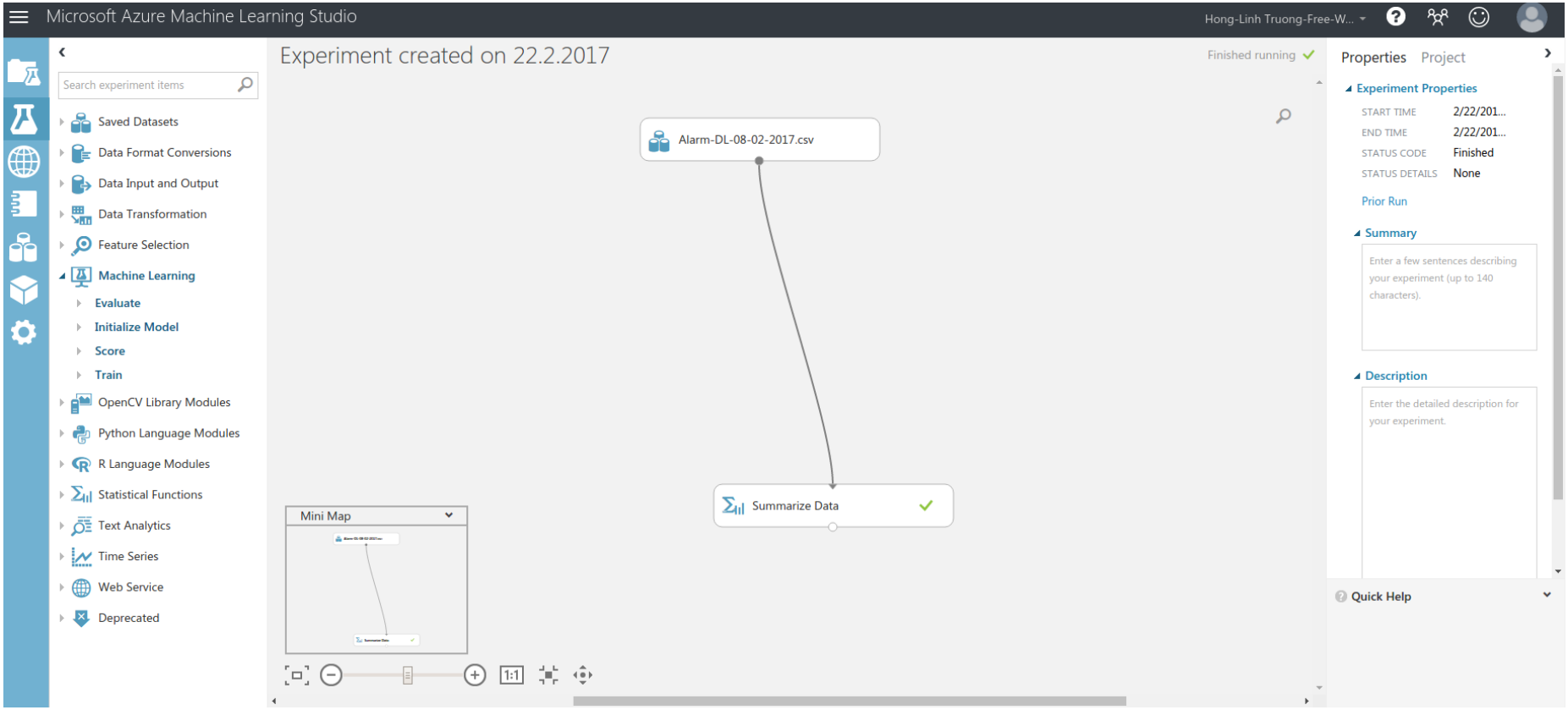
https://www.influxdata.com

Focus on time series data

- Collect
- Storage
- Visualize
- ETL



Machine Learning Stack



Emerging computing infrastructure/platform provisioning models– properties and issues

Some properties

- Rich types of services from multiple providers
  - Better choices in terms of functions and costs
- Concepts are similar but diverse APIs
- Strong dependencies/tight ecosystems

Some issues

- On-demand information management from multiple sources
- APIs complexity and API management
- Cross-vendor integration
- Execution in Multi-cloud environments
- Data locality

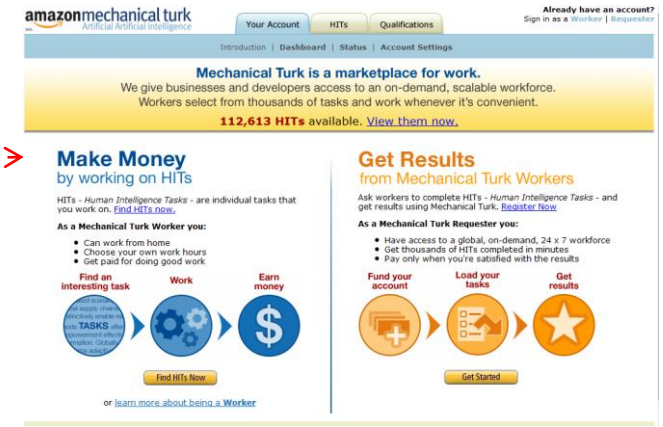
Emerging human computation models

- Crowdsourcing platforms
  - (Anonymous) people computing capabilities exploited via task bids
- Expert as Individual Compute Unit
  - An individual is treated like „a processor“ or “functional unit“. A service can wrap human capabilities to support the communication and coordination of tasks
- A set of individuals as *collectives*
  - A set of people and software that are initiated and provisioned as a service for solving tasks

The main point: humans are a computing element

Examples of human computation (2)

```
1  import edu.umass.cs.automan.adapters.MTurk._
2
3  object SimpleProgram extends App {
4    val a = MTurkAdapter { mt =>
5      mt.access_key_id = "XXXX"
6      mt.secret_access_key = "XXXX"
7    }
8
9    def which_one() = a.RadioButtonQuestion { q =>
10     q.budget = 8.00
11     q.text = "Which one of these does not belong?"
12     q.options = List(
13       a.Option('oscar, "Oscar the Grouch"),
14       a.Option('kermit, "Kermit the Frog"),
15       a.Option('spongebob, "Spongebob Squarepants"),
16       a.Option('cookie, "Cookie Monster"),
17       a.Option('count, "The Count")
18     )
19   }
20
21   println("The answer is " + which_one()())
22 }
```



Source: Daniel W. Barowy, Charlie Curtsinger, Emery D. Berger, Andrew McGregor: **AutoMan: a platform for integrating human-based and digital computation**. OOPSLA 2012: 639-654

Human computation models – properties and issues

Some properties

- Huge number of people
- Capabilities might not know in advance
- Unpredictable behavior
- Simple coordination models

Some issues

- Reliability
- Quality control
- Reliability assurance
- Proactive, on-demand acquisition
- Incentive strategies
- Collectives

Emerging Software-defined *

- Software-defined concepts
  - To have better way to manage dynamic changes in computation, network and data
  - Capabilities to manage and control computation, data, and network features at runtime using software
  - Management and control are performed via open APIs
- Software-defined techniques
  - Software-defined networking, Software-defined storage, Software-defined services

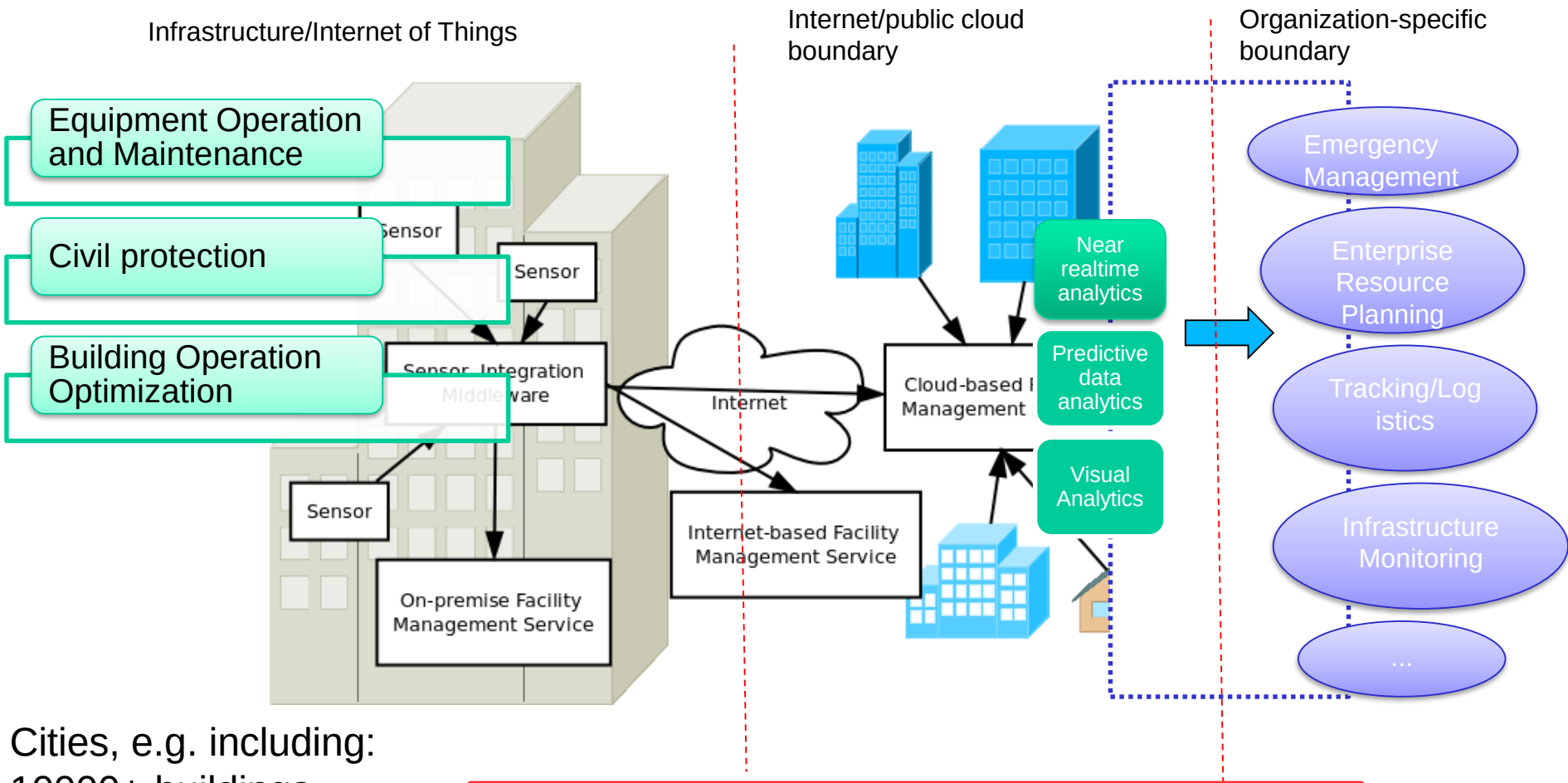
- K IRKPATRICK , K. Software-defined networking. Commun. ACM 56, 9 (Sept. 2013), 16–19.
- L ANGO , J. Toward software-defined slas. Commun. ACM 57, 1 (Jan. 2014), 54–60.
- S UGIKI , A., AND K ATO , K. Elements and composition of software-defined data centers. In Proceedings of the Posters and Demo Track (New York, NY, USA, 2012), Middleware '12, ACM, pp. 3:1–3:2.

Discussion time:

DO I NEED TO STUDY THEM ALL? WHY?

USE CASES/SCENARIOS

Smart building management



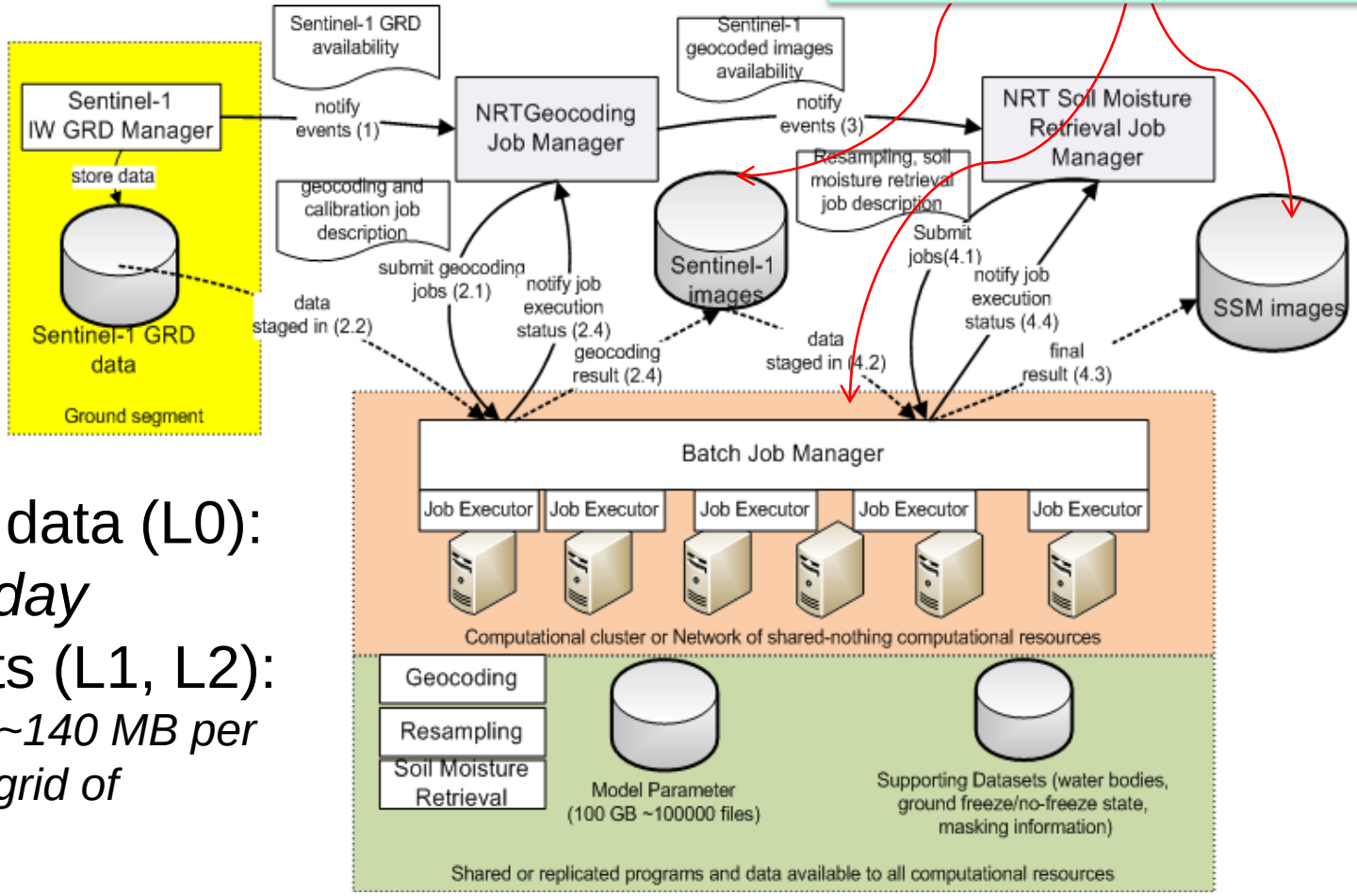
Cities, e.g. including:
10000+ buildings
1000000+ sensors

Can we combine open government data with building monitoring data?

Earth Observation

Can we combine them with open government data?

Soil moisture analysis for Sentinel-1

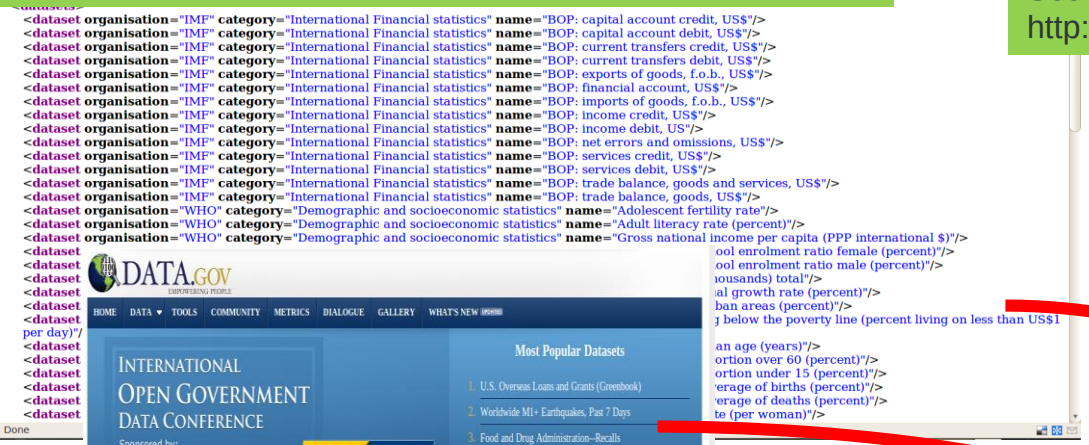


A lot of input data (L0):
~2.7 TB per day
A lot of results (L1, L2):
e.g., L1 has *~140 MB per day for a grid of 1kmx1km*

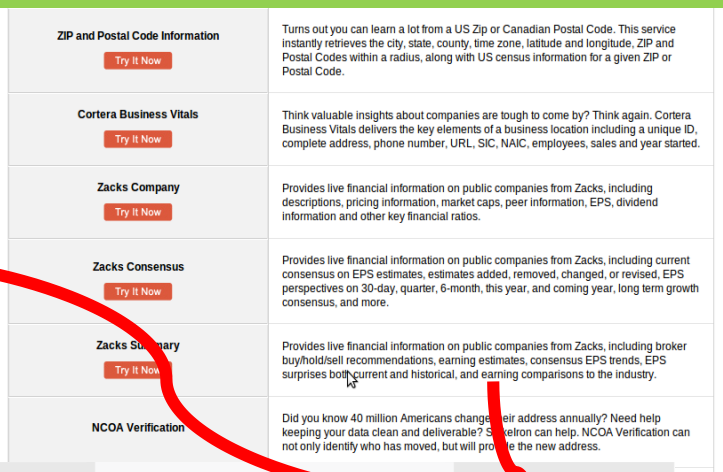
Michael Hornacek,Wolfgang Wagner, Daniel Sabel, Hong-Linh Truong, Paul Snoeij, Thomas Hahmann, Erhard Diedrich, Marcela Doubkova, **Potential for High Resolution Systematic Global Surface Soil Moisture Retrieval Via Change Detection Using Sentinel-1**, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, April, 2012

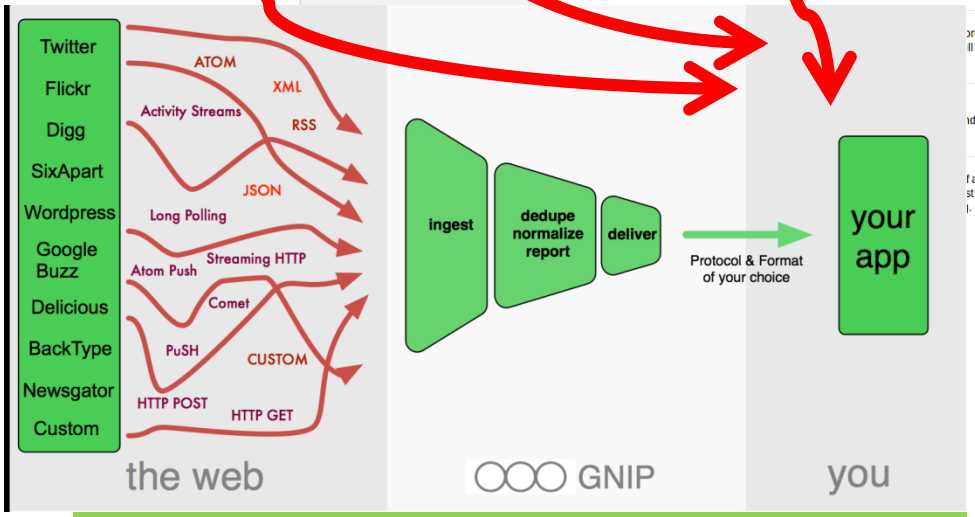
Open data and social data

Source: http://www.undata-api.org/



Source: http://www.strikeiron.com/Catalog/StrikeIronServices.aspx





Source: http://docs.gnip.com/w/page/23722723/Introduction-to-Gnip

Video analytics + business applications/public security

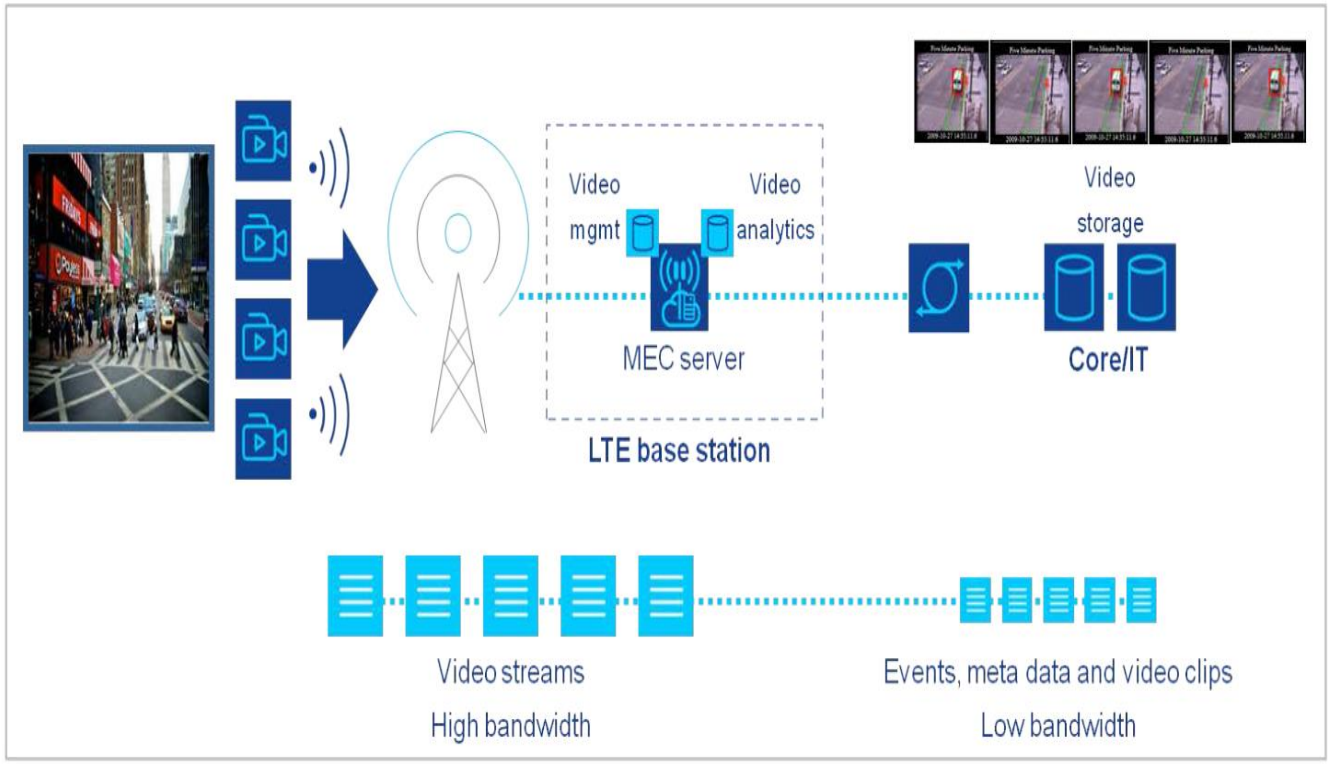


Figure source:
https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing-introductory_technical_white_paper_v1%2018-09-14.pdf

Critical infrastructures/services for citizens and business

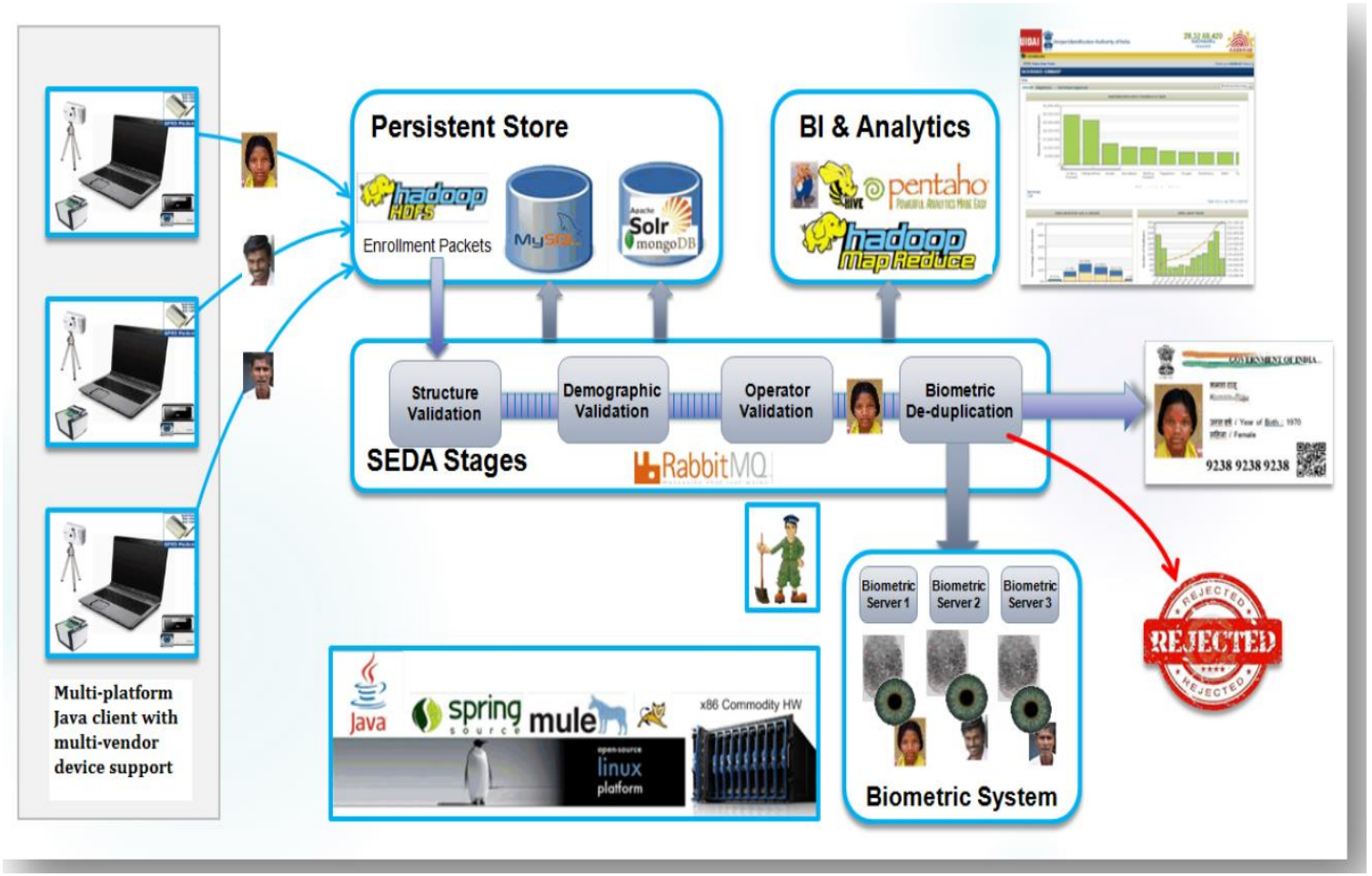
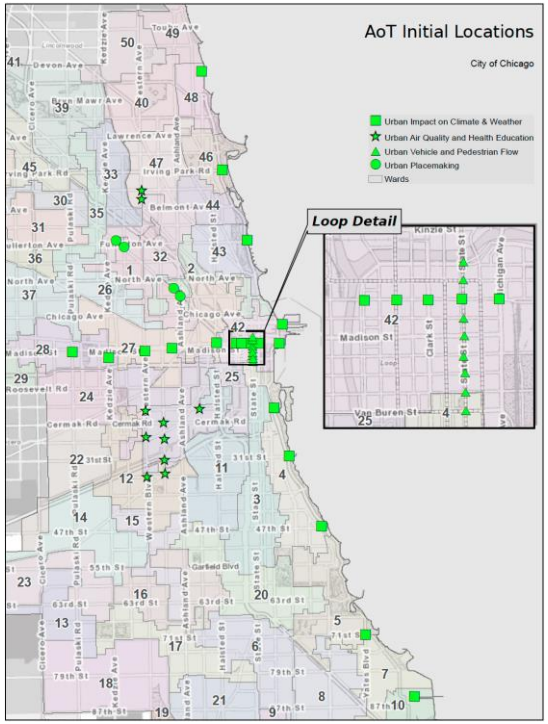
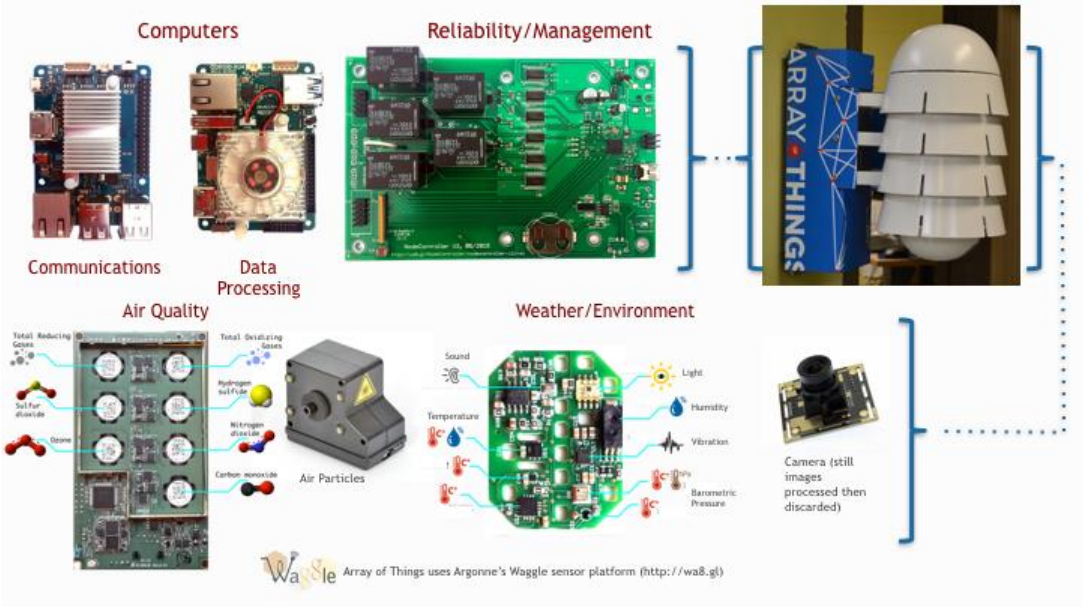


Figure source: http://uidai.gov.in/images/AadhaarTechnologyArchitecture_March2014.pdf

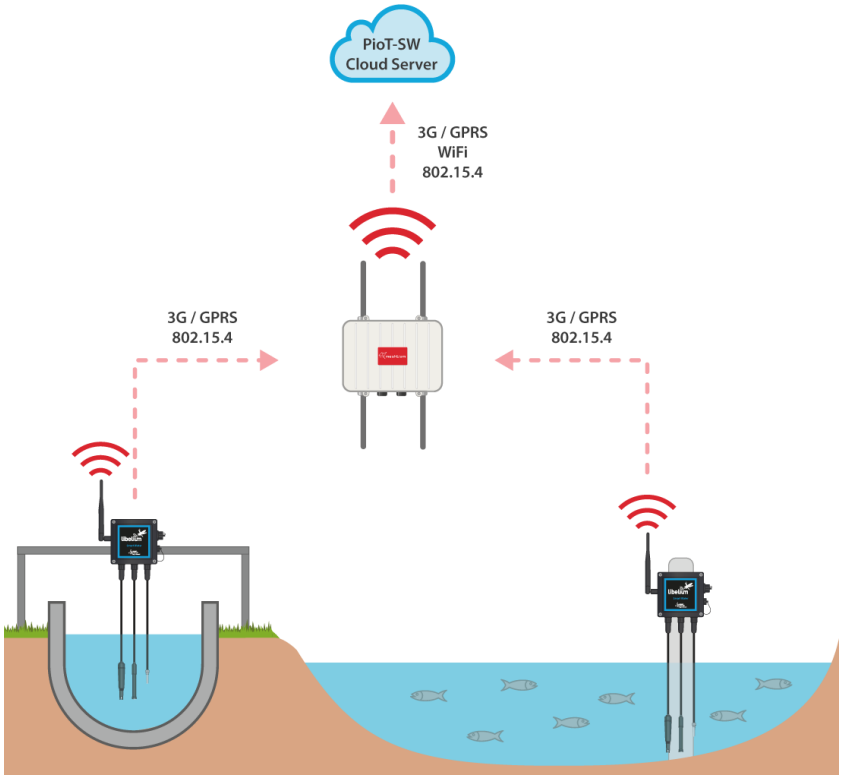
IoT data in Cities

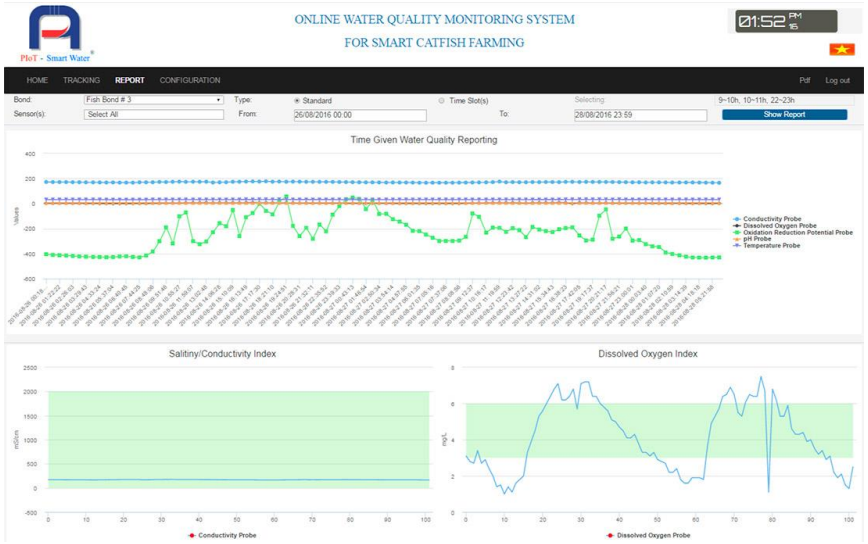




https://arrayofthings.github.io/node.html

Smart Farming





Source: http://www.libelium.com/fish-farm-monitoring-in-vietnam-by-controlling-water-quality-in-ponds-and-tanks/

Geo Sports

Geo Sports: Picture courtesy Future Position X, Sweden

sensing → analytics → Decision

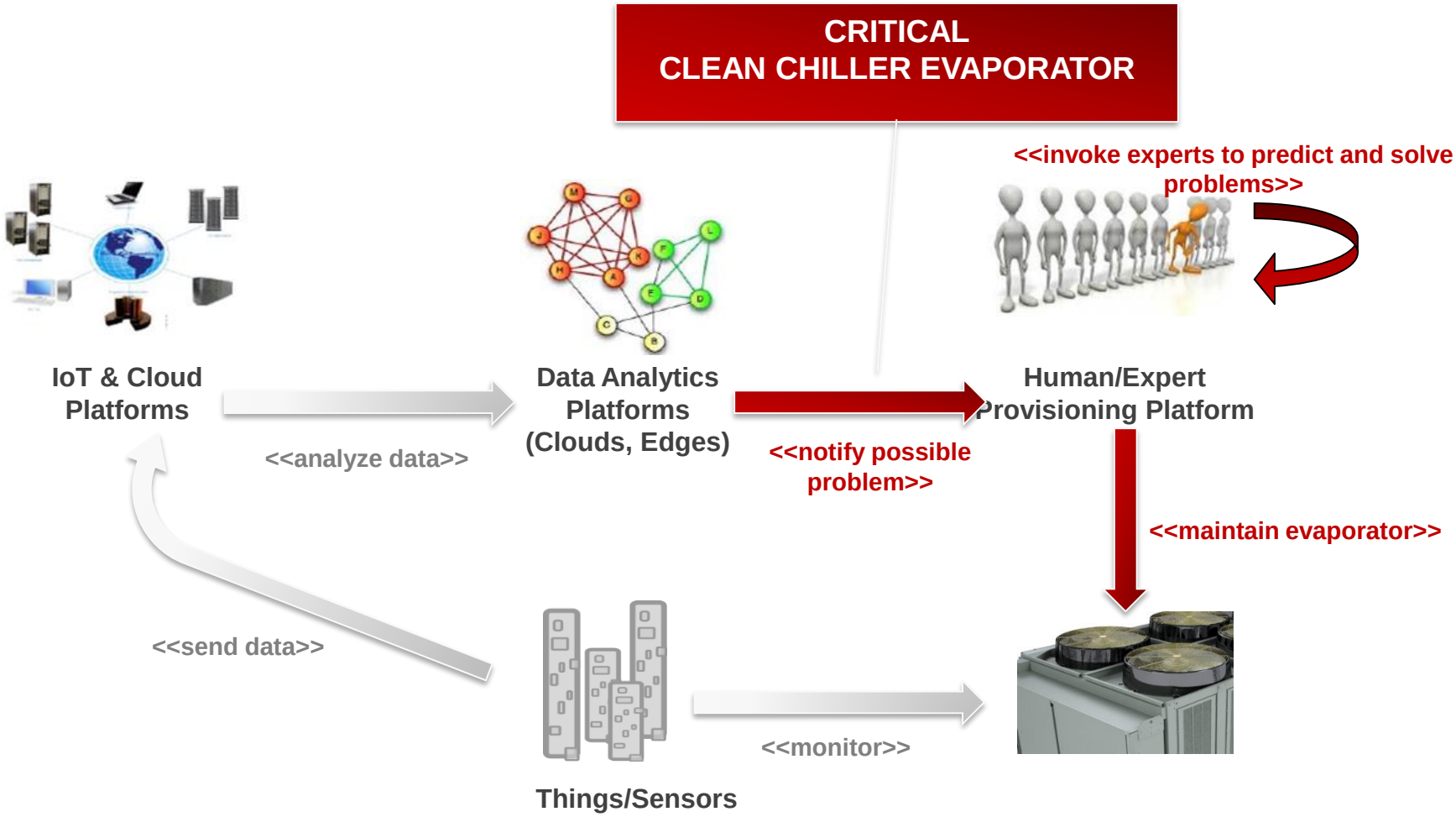
Drones for logistics

Figure 16: Urban First and Last Mile

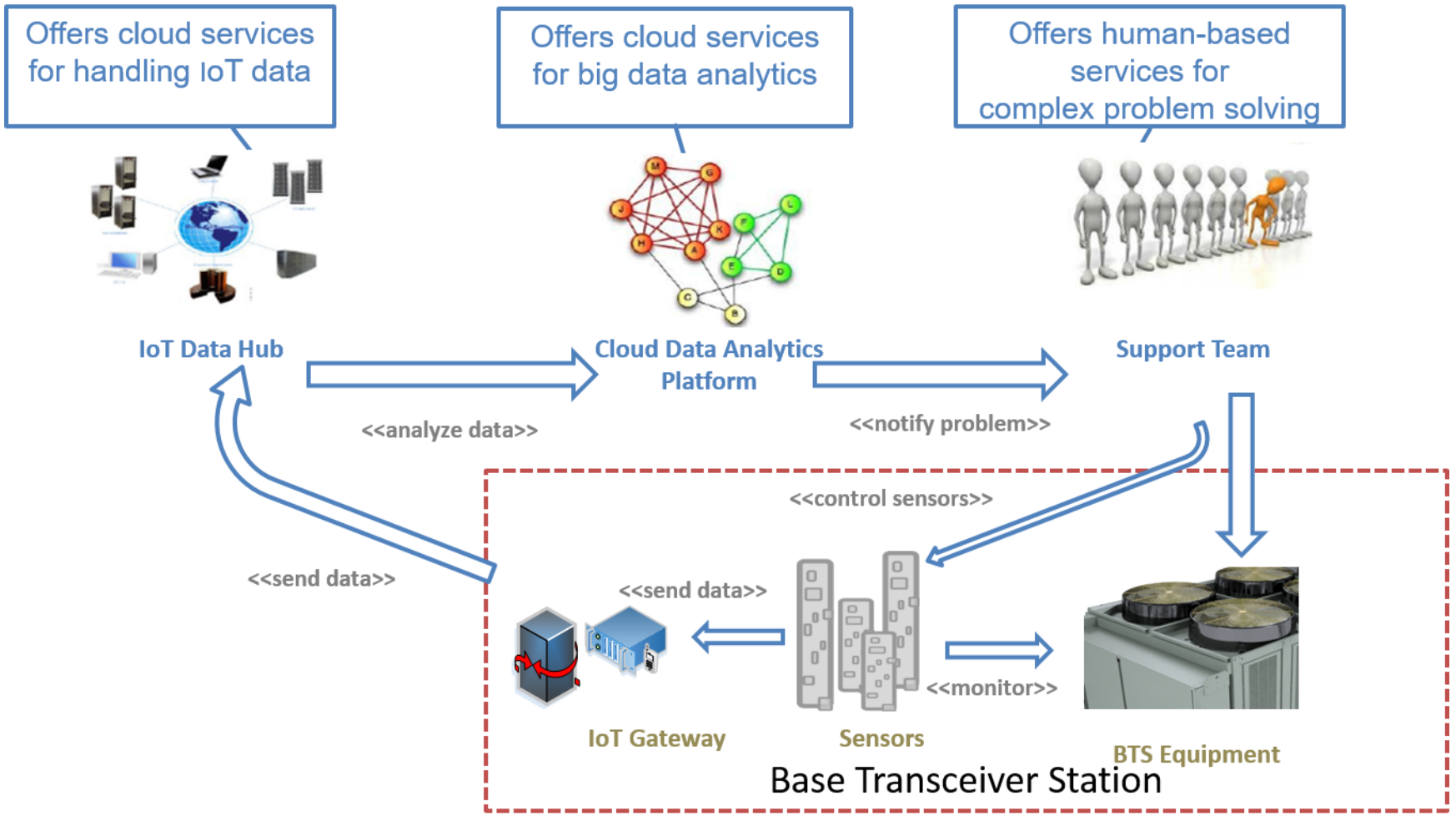
Source: DHL Trend Report “Unmanned Aerial Vehicles”
http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/dhl_trend_report_uav.pdf

CONVERGENCE OF MULTIPLE COMPUTING MODELS

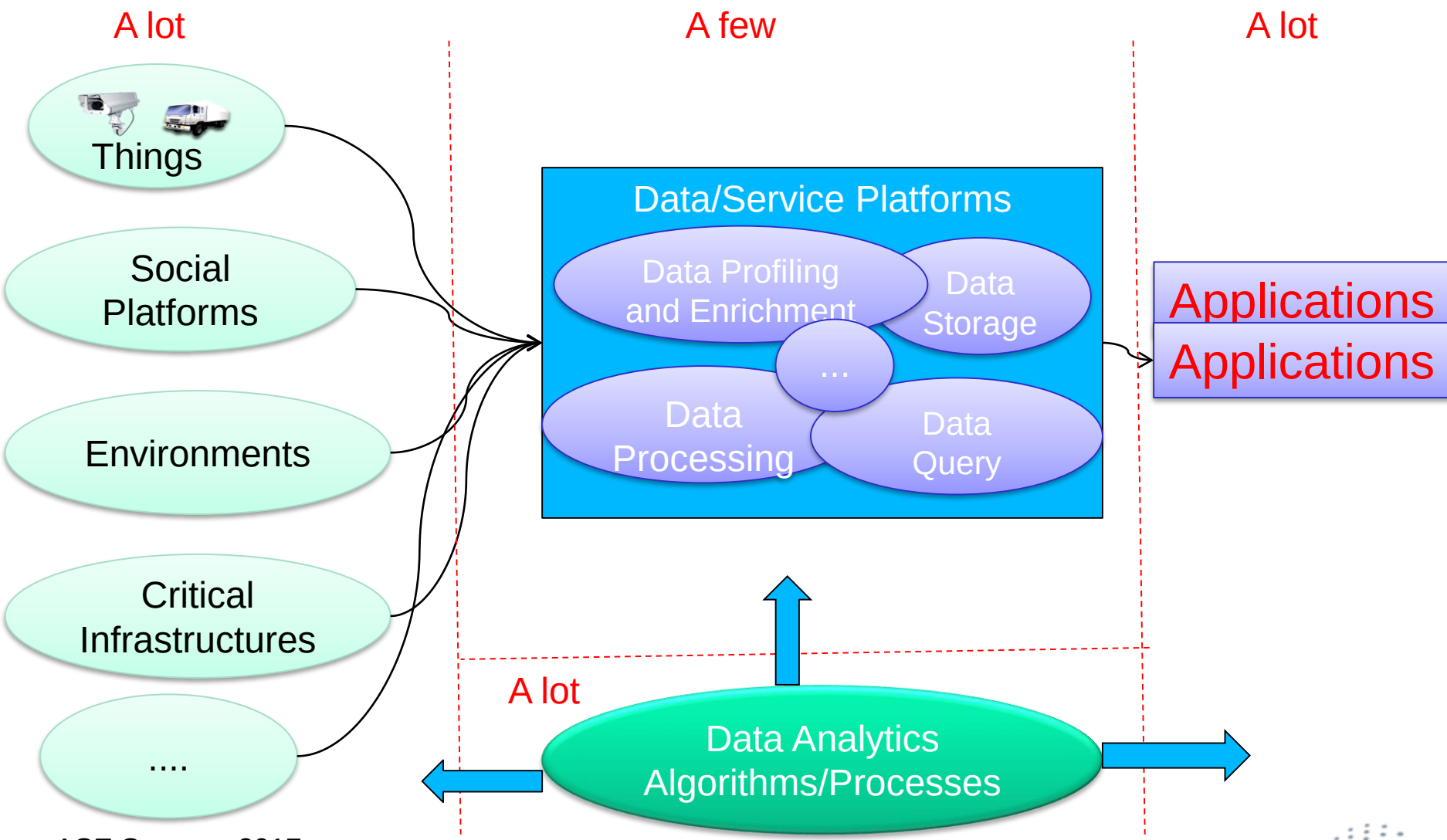
Predictive Maintenance in Cities



Predictive Maintenance in Telcos



Cloud-based Analytics

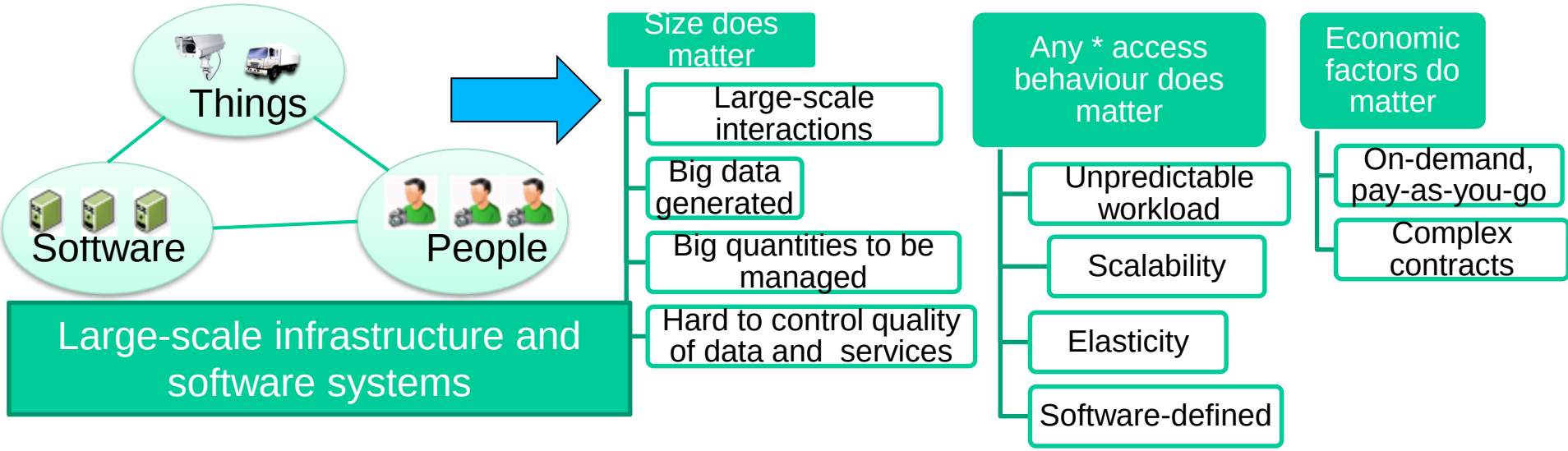


BRING YOUR OWN EXPERIENCE: CLOUD-BASED ANALYTICS

See also http://www.allthingsdistributed.com/2015/03/the-importance-of-cloud-based-analytics.html

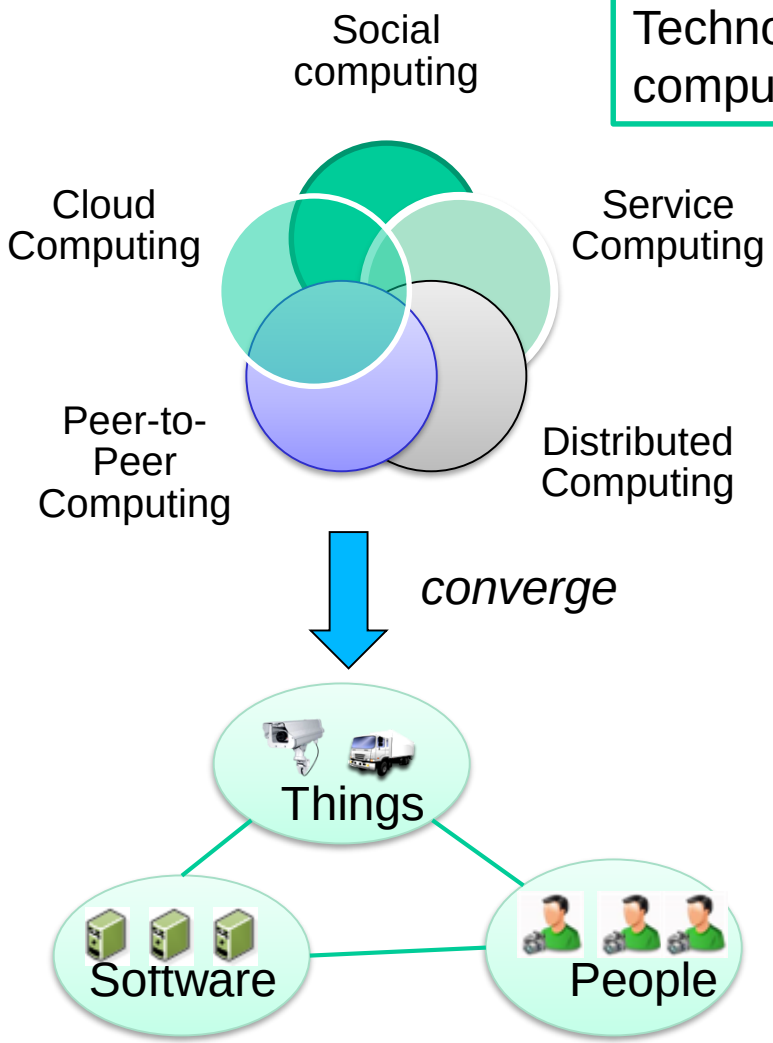
Today‘s Computing Models

- Internet infrastructure and software connect *contents, things, and people,* each has different roles (*computation, sensing, analytics, etc.*)



Today's Computing Models

Technologies and computing models

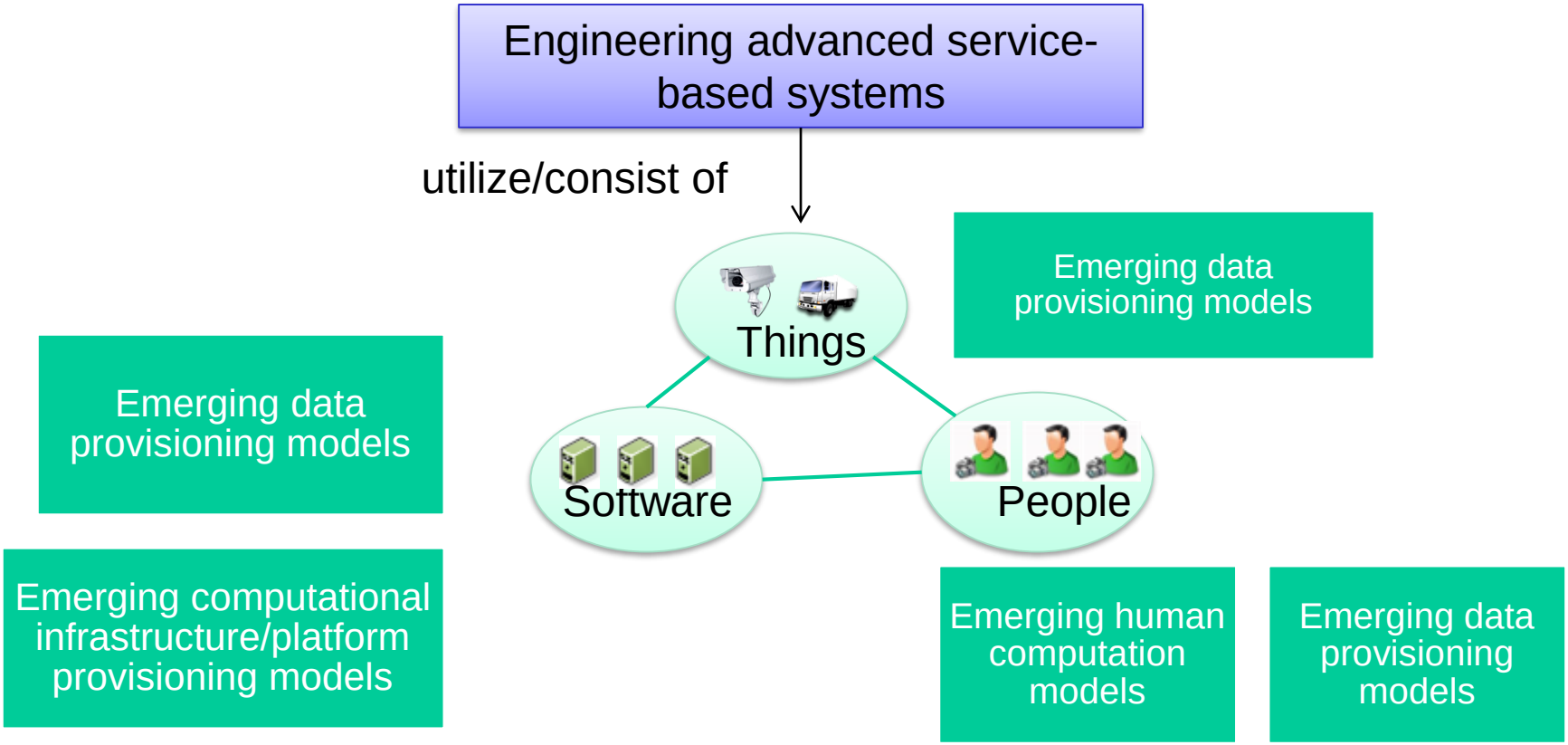


- Big and high performance centralized data analytics
- IoT data streaming analytics
- Large-scale applications spanning data centers and edge servers/gateways
- Adaptive collective systems of humans and machines

introduces

Emerging forms of computing models, systems and applications

Summary of emerging models wrt advanced service-based systems



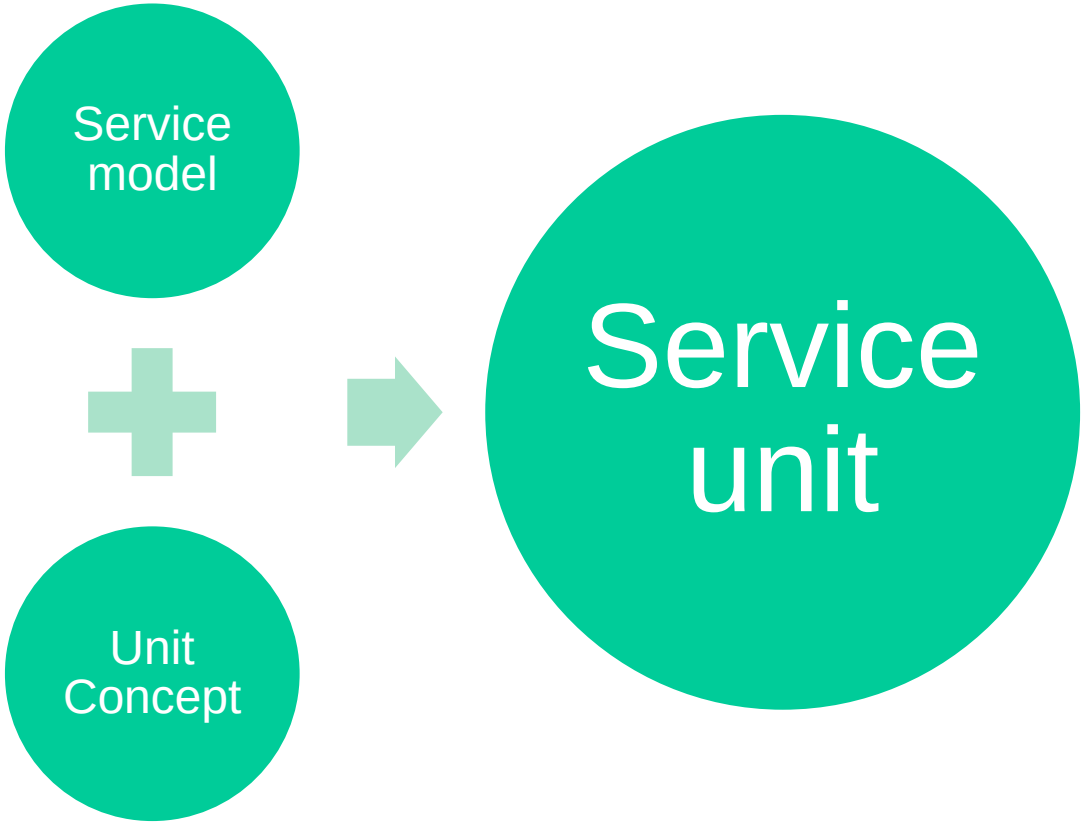
Challenges in Virtualization, Programming, Communication, and Coordination, etc.

ADVANCED SERVICES ENGINEERING‘S FOCUS

Single service/platform engineering(1)

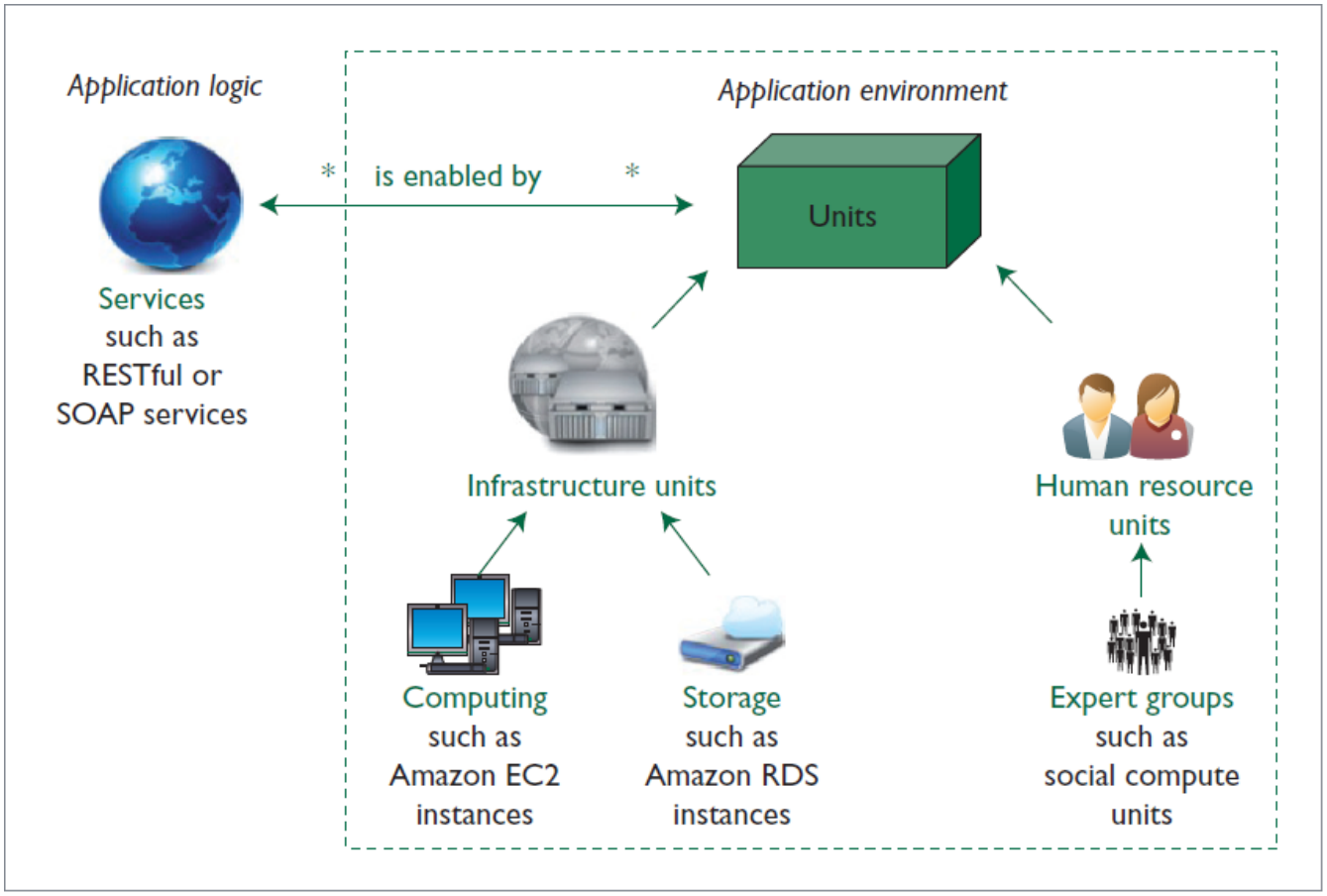
- The service model can be applied to things, people and software

Consumption, ownership, provisioning, price, etc.



„basic component“/“basic function“ modeling and description

Single service/platform engineering – service units (2)



Source: Stefan Tai, Philipp Leitner, Schahram Dustdar: Design by Units: Abstractions for Human and Compute Resources for Elastic Systems. IEEE Internet Computing 16(4): 84-88 (2012)

Single service/platform engineering – service unit provisioning

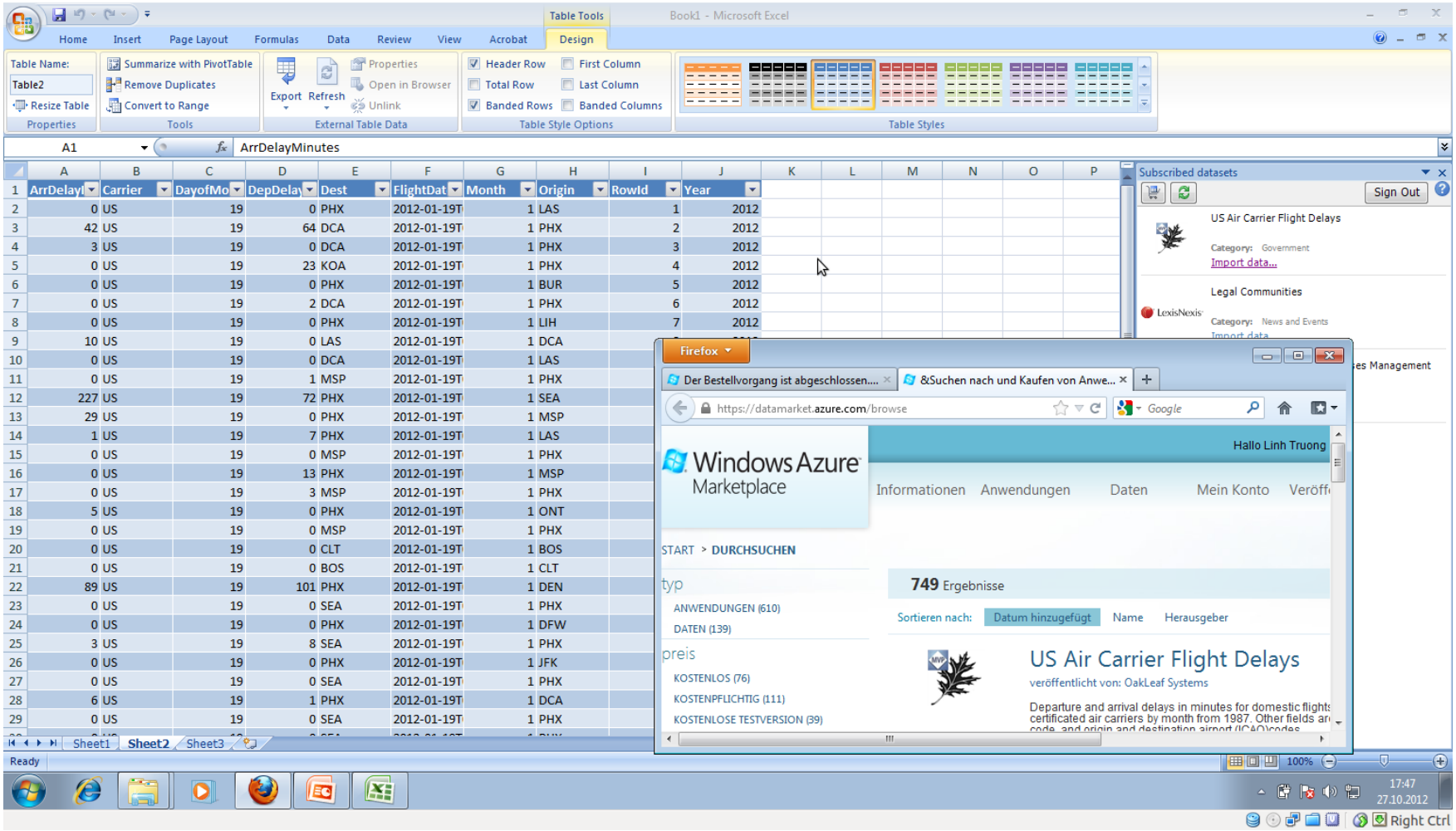
- Provisioning software under services
- Provisioning things under services
- Provisioning human under services
  - Crowd platforms of massive numbers of individuals
  - Collectives of humans

1. Mark Turner, David Budgen, and Pearl Brereton. 2003. **Turning Software into a Service**. *Computer* 36, 10 (October 2003), 38-44. DOI=10.1109/MC.2003.1236470 http://dx.doi.org/10.1109/MC.2003.1236470
2. Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. **The Internet of Things: A survey**. *Comput. Netw.* 54, 15 (October 2010), 2787-2805. DOI=10.1016/j.comnet.2010.05.010 http://dx.doi.org/10.1016/j.comnet.2010.05.010
3. Dominique Guinard, Vlad Trifa, Stamatis Karnouskos, Patrik Spiess, Domnic Savio: **Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services**. IEEE T. Services Computing 3(3): 223-235 (2010)
4. Schahram Dustdar, Kamal Bhattacharya: **The Social Compute Unit.** IEEE Internet Computing 15(3): 64-69 (2011)
5. Hong-Linh Truong, Schahram Dustdar, Kamal Bhattacharya "**Programming Hybrid Services in the Cloud**", Springer-Verlag, 10th International Conference on Service-oriented Computing (ICSOC 2012), November 12-16, 2012, Shanghai, China

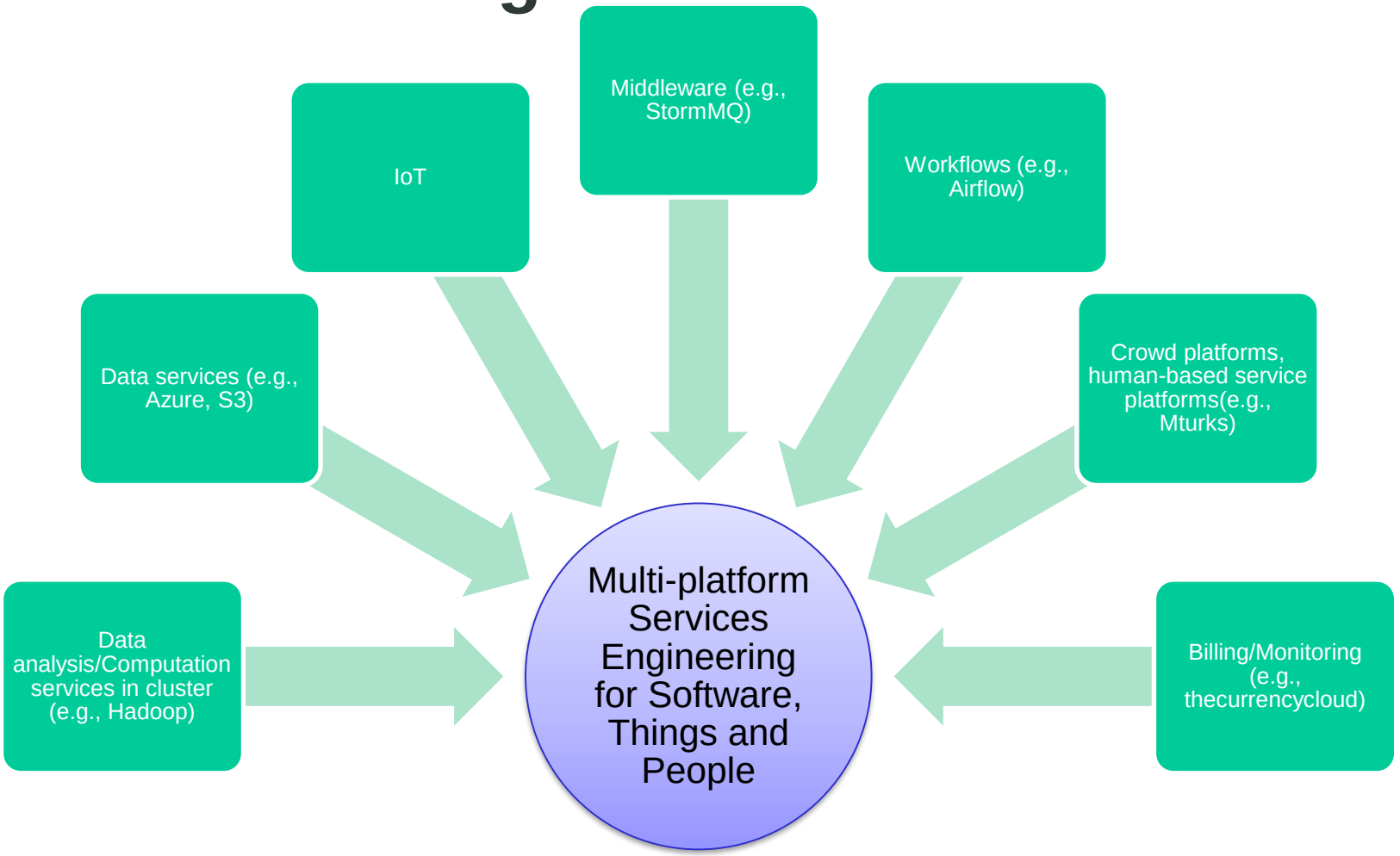
Single service/platform engineering – examples (1)

- Service engineering with a single system/platform
  - Using Excel to access Azure datamarket places
  - Using Boto to access data in Amazon S3
  - Using Hadoop/Spark within a cluster to analyze data
  - Using workflows to process data (e.g., Trident/Taverna/Airflow)
  - Using StormMQ/Kafka to store messages

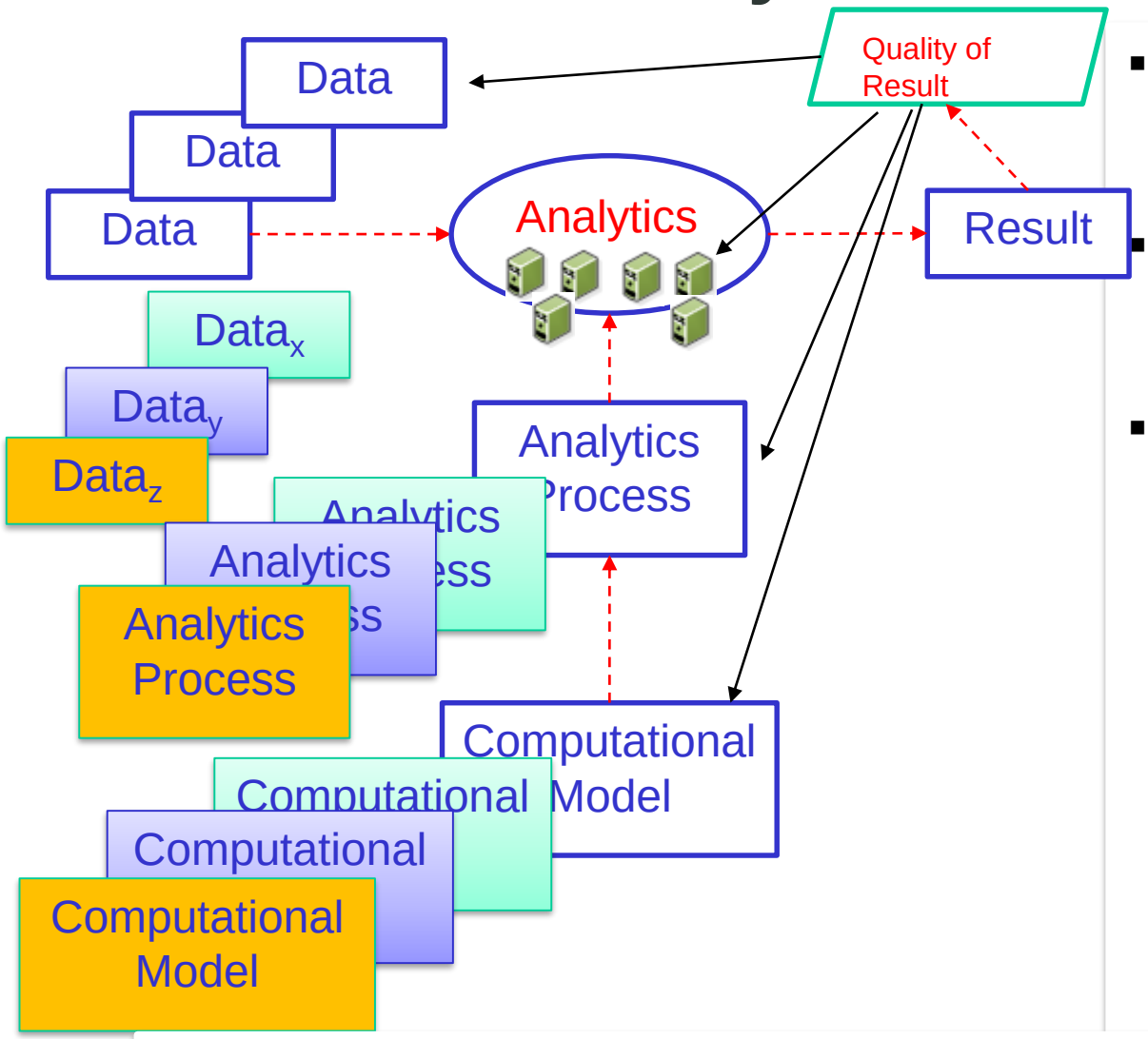
Single service/platform engineering – examples (2)



Internet-scale multi-platform services engineering – required technologies



Internet-scale service engineering – the elasticity



- **More data** → more computational resources (e.g. more VMs)
- **More types of data** → more computational models → more analytics processes
- Change **quality of analytics**
  - Change quality of data
  - Change response time
  - Change cost
  - Change types of result (form of the data output, e.g. tree, visual, story, etc.)

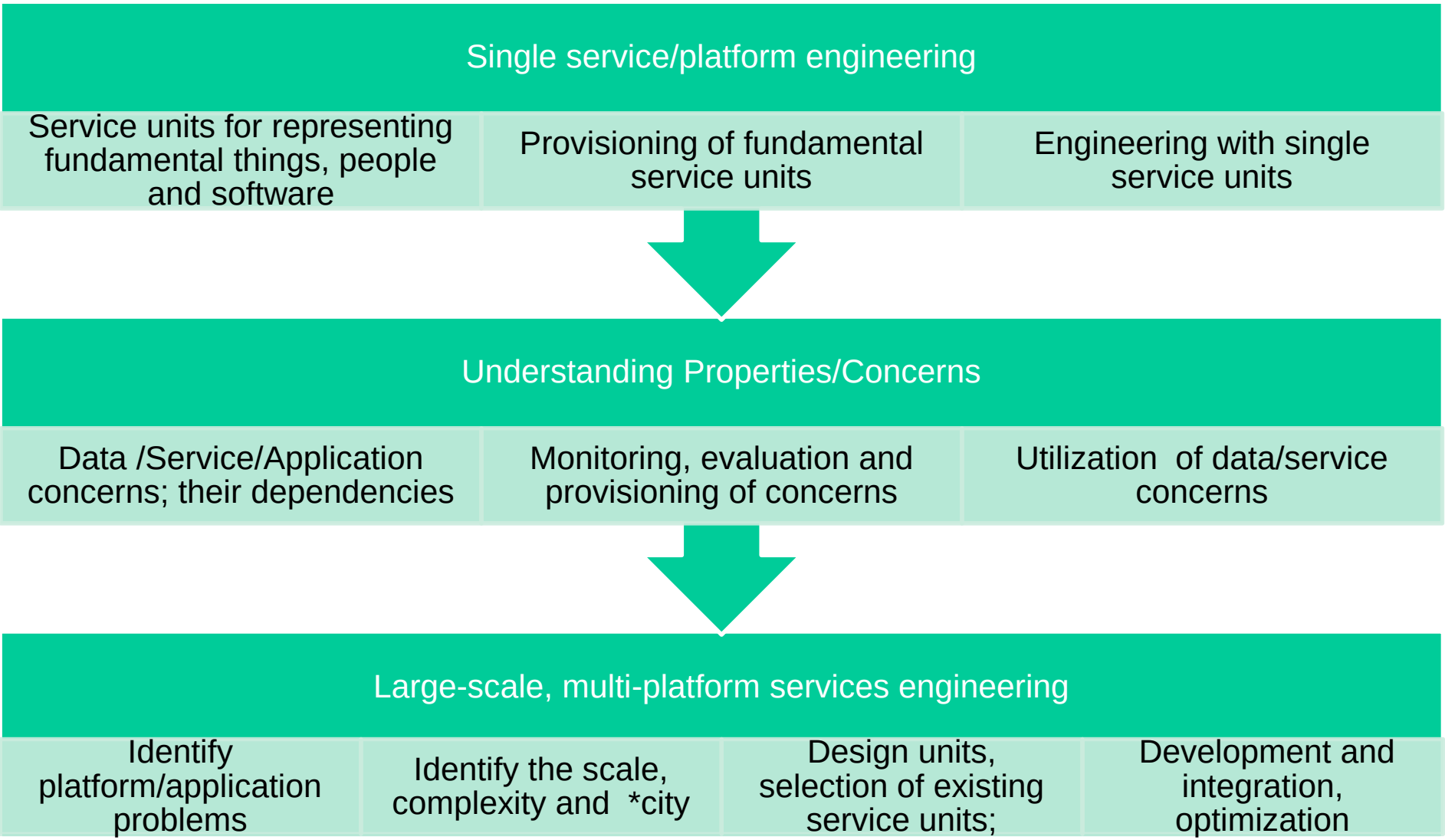
A

Hong-Linh Truong, Schahram Dustdar, "Principles of Software-defined Elastic Systems for Big Data Analytics", (c) IEEE Computer Society, IEEE International Workshop on Software Defined Systems, 2014 IEEE International Conference on Cloud Engineering (IC2E 2014), Boston, Massachusetts, USA, 10-14 March 2014

Internet-scale service engineering -- big/near-real time data impact

- Which are data concerns that impact the data processing?
- How to use data concerns to optimize data analytics and service provisioning?
- How to use available data assets for advanced services in an elastic manner?
- What are the role of human-based servies in dealing with complex data analytics?

Internet-scale service engineering -- Steps



Exercises

- Read papers mentioned in slides
  - Get their main ideas
- Check services mentioned in examples
  - Examine capabilities of the mentioned services
    - Including price models and underlying technologies
  - Examine their size and scale
  - Examine their ecosystems and dependencies
- Work on possible categories of single service units that are useful for your work
  - Some common service units with capabilities and providers

Thanks for your attention

Hong-Linh Truong
Distributed Systems Group, TU Wien
truong@dsg.tuwien.ac.at
http://dsg.tuwien.ac.at/staff/truong
@linhsolar