Advancements in Next Generation Computing and Communications in the IoT age, Faculty Development Program, Sister Nivedita University

12th July 2025

Next Generation Computing in the Internet-of-Things Age

Professor Dr. rer. nat. habil. Sven Groppe
https://www.ifis.uni-luebeck.de/~groppe

Stations of my academic life

Research Areas

- Artificial Intelligence, Machine Learning and Data Science
  - LLMs, Agentic Workflows, Mathematical Optimizations, Graph Neural Networks, Chatbots, Reasoning
- Data Management Tasks
  - Query Processing & Opt., Indexing, Mapping, Compression, Replication, Caching, Transaction Handling
- Data Models
  - Knowledge Graphs, Semantic Web, Property Graphs, Relational Data, XML
- Types of Data
  - Big Data, Data Streams
- Emergent Hardware Technologies
  - Many-Core CPU, GPU, FPGA, Quantum Computer
- Platforms
  - Internet, Internet of Things, Cloud, Post-Cloud (Fog/Edge/Dew Computing), P2P, Mobile, Parallel and Main Memory Servers
- Advanced Applications
  - Citizen Science, Customer Communications, Pandemics like Covid-19, Software Vulnerability Prediction
- Sustainability
  - Sustainable Computing/AI, Applications for Sustainability

Lectures by Sven Groppe

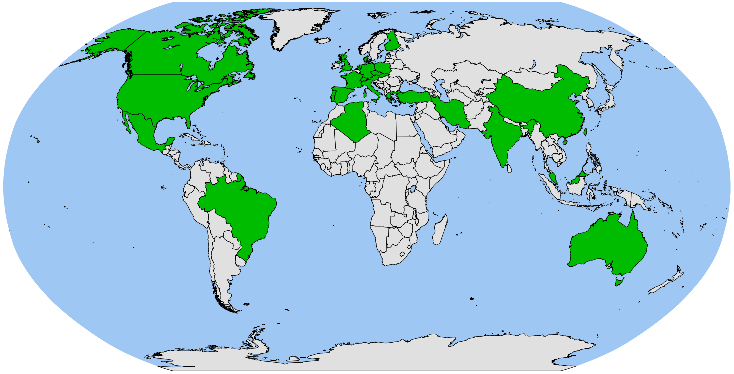
52 lectures in the areas of

- Information Systems N T (inclusive Knowledge Graphs, LLMs and Graph Neural Networks) (since 2024)
- Quantum Computing N T (inclusive Quantum Machine Learning) (since 2022)
- Semantic Web N T (since 2011)
- Databases (2014, since 2024)
- Mobile and Distributed Databases N (since 2008)
- Cloud and Web Technologies N T (since 2015)
- Algorithms and Datastructures (2014)
- XML Databases N (2007)
- Compiler Design N T (2006/2007)
- Next Web Generation (2006, together with M. Zaremba)

N Newly designed T Online Tutorials

Supervision and Publication Record

- 3 supervised dissertations, 6 current PhD students
- $\approx$ 100 bachelor/master/diploma thesis/student projects
- $>$ 200 publications
  - 16 publications at A/A1 ranked conferences[1]
  - 206 co-authors affiliated with organizations in 28 countries on 6 continents



  - 50 publications (25%) are co-authored by bachelor/master students (being students at time of writing)

[1] according to Conference Ranks

Project Grants (≈ 2M €)

1. D I High Quality Knowledge Graphs from recent English, French and German Emergent Trends with the example of COVID-19 (2022-2025)
2. B QC4DB: Accelerating Relational Database Management Systems via Quantum Computing (2022-2025)
3. D Hybrid^2-Indexstrukturen für Hauptspeicherdatenbanken (2019-2024)
4. D BigSIoT: Big Data Management for the Semantic Internet of Things (2020-2023)
5. D Hardwarebeschleunigung von Semantic Web Datenbanken durch dynamisch rekonfigurierbare FPGAs (2013-2015)
6. W Beschleunigung relationaler Datenbanken mittels laufzeitadaptiver FPGA-Cluster (2013-2015)
7. D Logisch und Physikalisch Optimierte Semantic Web Datenbank-Engine (2007-2009)

D DFG projects I International Project W BMWi/ZIM B BMBF

Scientific Services

- General Chair
  - The International Conference on Applied Machine Learning and Data Analytics (AMLDA) '23
  - International Semantic Intelligence Conference (ISIC) ('21-'22)
  - International Health Informatics Conference (IHIC) ('22-'24)
  - International EdTech Conference (IEdTC) '23
- Workshop Chairs
  - Quantum Data Science and Management (QDSM)@VLDB ('23-'24)
  - Semantic Big Data (SBD)@SIGMOD ('16-'20)
  - Big Data in Emergent Distributed Environments (BiDEDE)@SIGMOD ('21-'23)
  - Very Large Internet of Things (VLIoT)@VLDB ('17-'22)
- many other scientific services
  - $\approx$ 141 PC memberships
  - reviewer of $\approx$ 42 journals
  - editor of 4 journals
  - ...

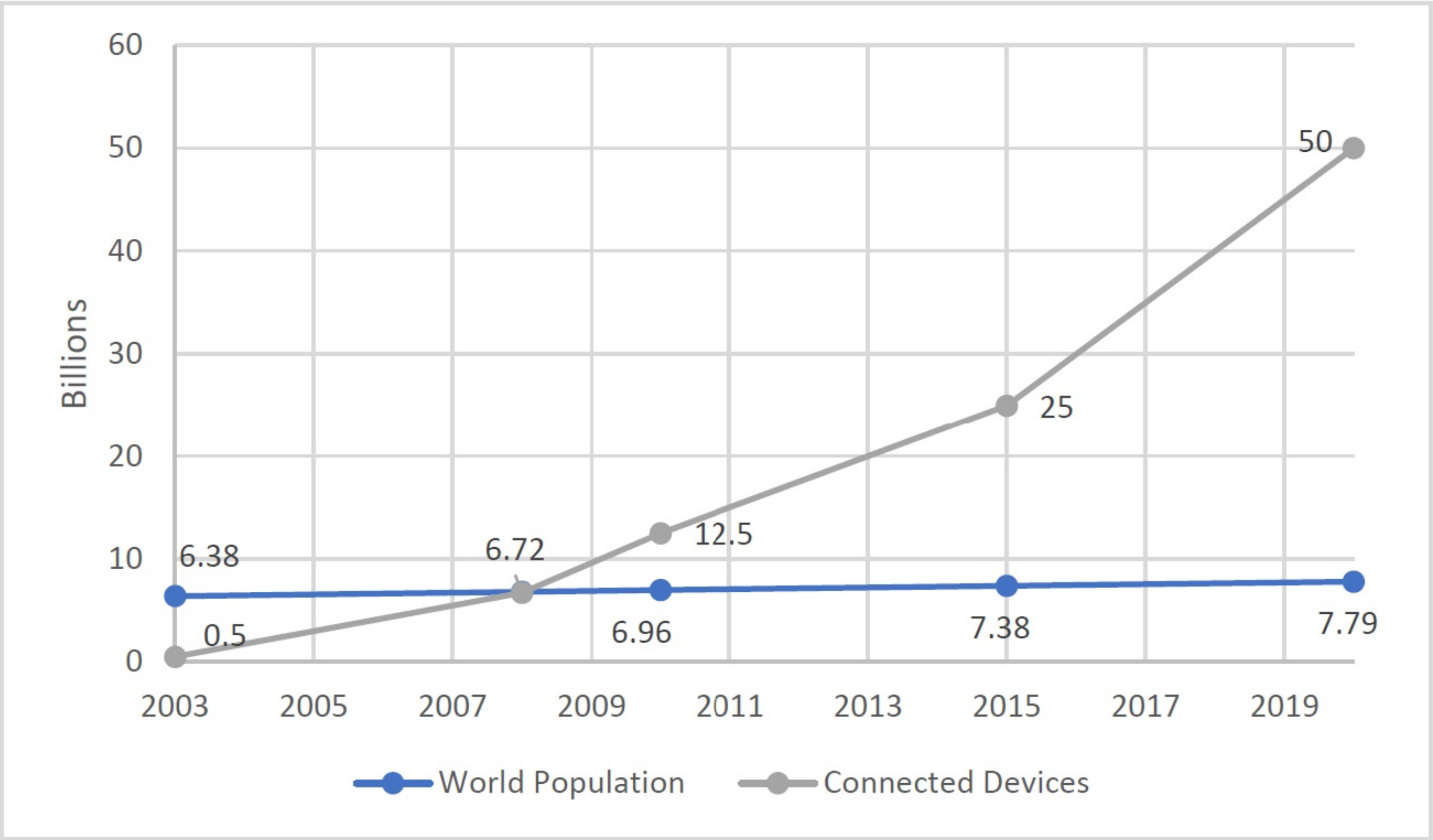
Definitions of Internet-of-Things (IoT)

- Internet is extended by adding things as network nodes.
  - connect those physical objects that are not yet connected with the help of the Internet and
  - thus create significant added value[1]
    - a better collaboration between humans and machines through new types of applications in various areas.
    - Application domains like Smart City, Smart Healthcare, Smart Farming, Smart Factory, ...
- "IoT allows people and things to be connected Anytime, Anyplace, Anything and Anyone ideally using any path/network and Any service."[2]

[1] Hanes et al., IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. 1st ed. Cisco Press, 2017

[2] Guillemin, P. and Friess, P. Internet of Things: Strategic Research Roadmap. Tech. rep. Oct. 2009.

CISCO: Age of IoT emerged between 2008 and 2009



Things in IoT are heterogeneous

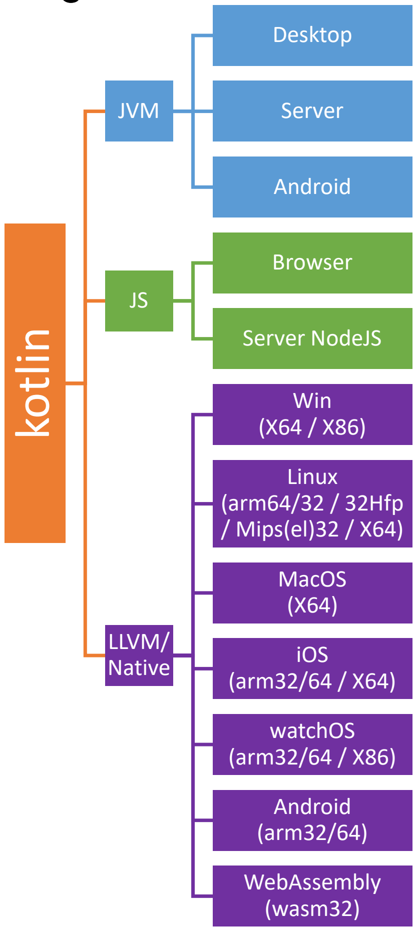
- Different kinds of devices: Smartphone/-watch-/tv/..., sensors, routers, tablets, raspberry pi (and similar), ...
- Hardware (e.g. CPU)
- Operating system (OS)
- Network protocols
- Capabilities
  - Performance of CPU
  - Storage capacity
  - Access to networks (wireless, stationary, ...)
  - ...
- ...

Multi-Platform Development (e.g. of DBMS)

- Native Binaries via C/C++
  - support of a new platform: porting code is necessary
  - code close to hardware, fast execution
  - direct access to native libraries
  - doesn't run in browser
  - most server DBMS: C/C++ code
- Java/Java Virtual Machine (JVM)
  - runs on many platforms (without porting code)
  - interpreted bytecode, via Just-In-Time compilation comparable speed to native execution
  - no direct access to native libraries
  - does neither run on iPhone nor in browser
  - many NoSQL/NewSQL/Cloud DBMS: Java (or JVM language like Scala) code
- Code generation for query processing via C/C++ or Janino-Compiler (JVM)

Multi-Platform Development with Kotlin

Targets:



- Most target platforms are supported
- Splitting the project in platform-**in**dependent and platform-dependent code
  - Platform-dependent code can be partly coded in the programming language of the target platform (e.g., Java for JVM, JS for Web)
- Enables one code repository for various target platforms
  - Sharing of code between server & (various) clients
- Avoids efforts to port code (into other programming languages)

Multi-Platform Development with Kotlin

- Common Module
  - Code independent of platforms containing declarations for platform dependent code without implementation, e.g.:

```
expect fun formatString(source: String, vararg args: Any): String
expect annotation class Test
```

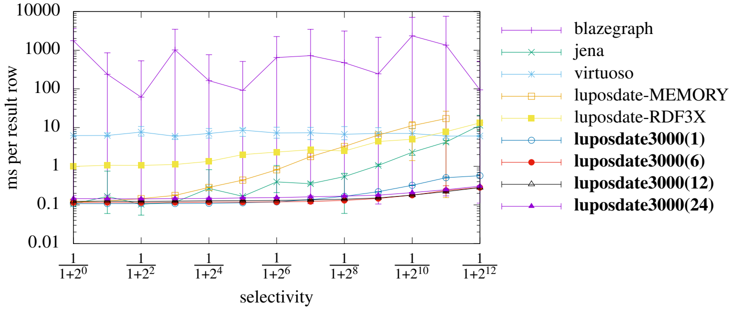
- Platform Module
  - Implementation of within the common module declared platform-dependent code (and other platform-dependent code), e.g.:

```
actual fun formatString(source: String, vararg args: Any) =
                                          String.format(source, args)
actual typealias Test = org.junit.Test
```

- Regular Module
  - depend on platform modules or platform modules depend on this module
- **However:** **High compilation times**, **faster**: Including different sets of source code directories for different targets and configurations (e.g., centralized, Cloud, P2P, browser, ...)

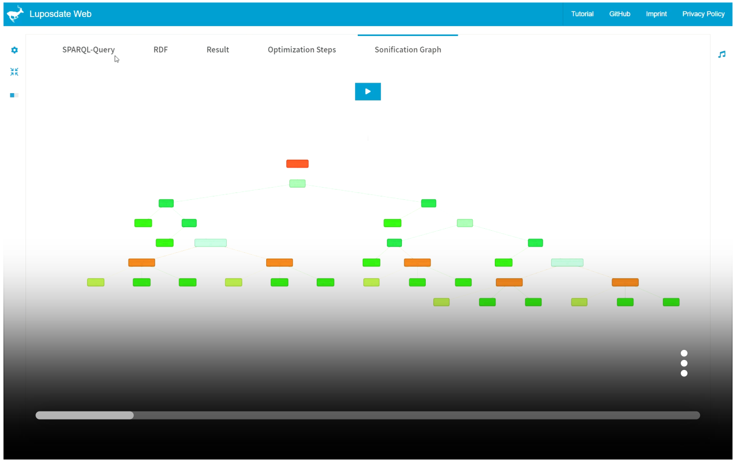
The Power of Multi-Platform: LUPOSDATE3000

- ultra-fast in jvm...



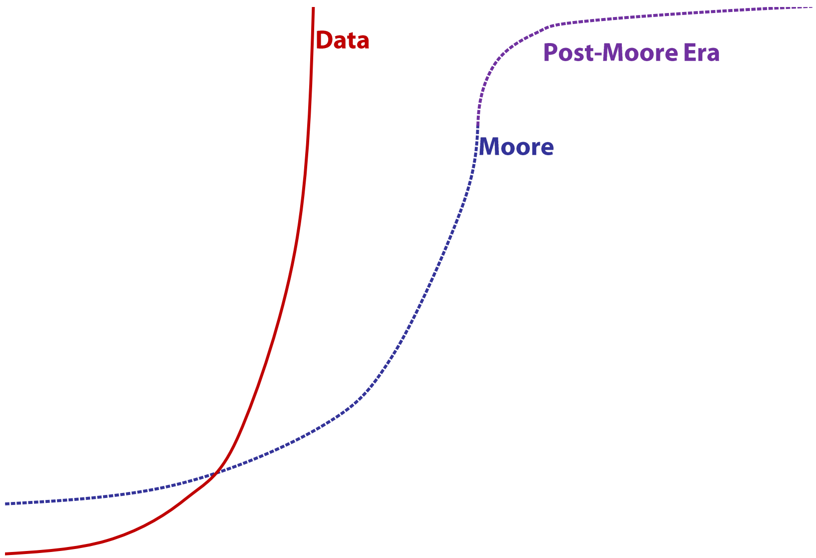
B. Warnke, M.W. Rehan, S. Fischer, S. Groppe: Flexible data partitioning schemes for parallel merge joins in semantic web queries in: BTW'21

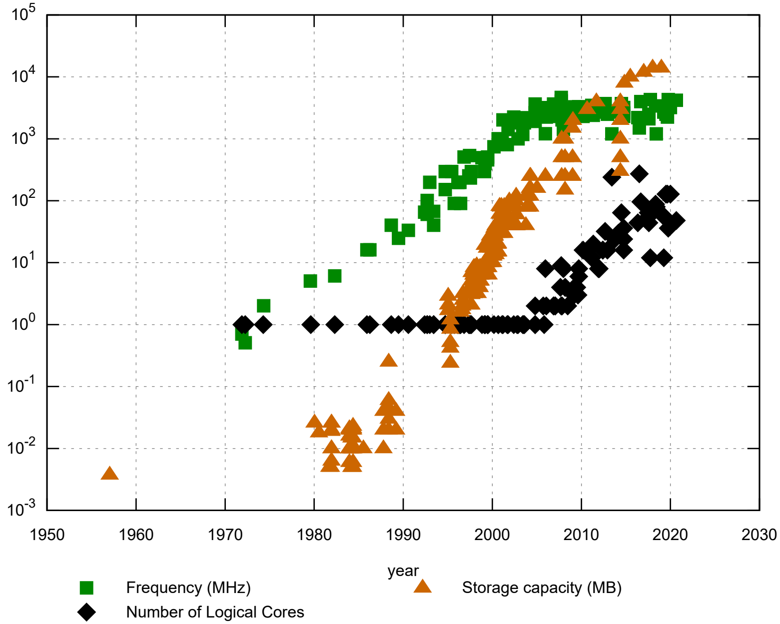
- ...but also enabling web demos running completely in the browser!



S. Groppe, R. Klinckenberg, B. Warnke. Sound of Databases: Sonification of a Semantic Web Database Engine. PVLDB, 14(12), 2021

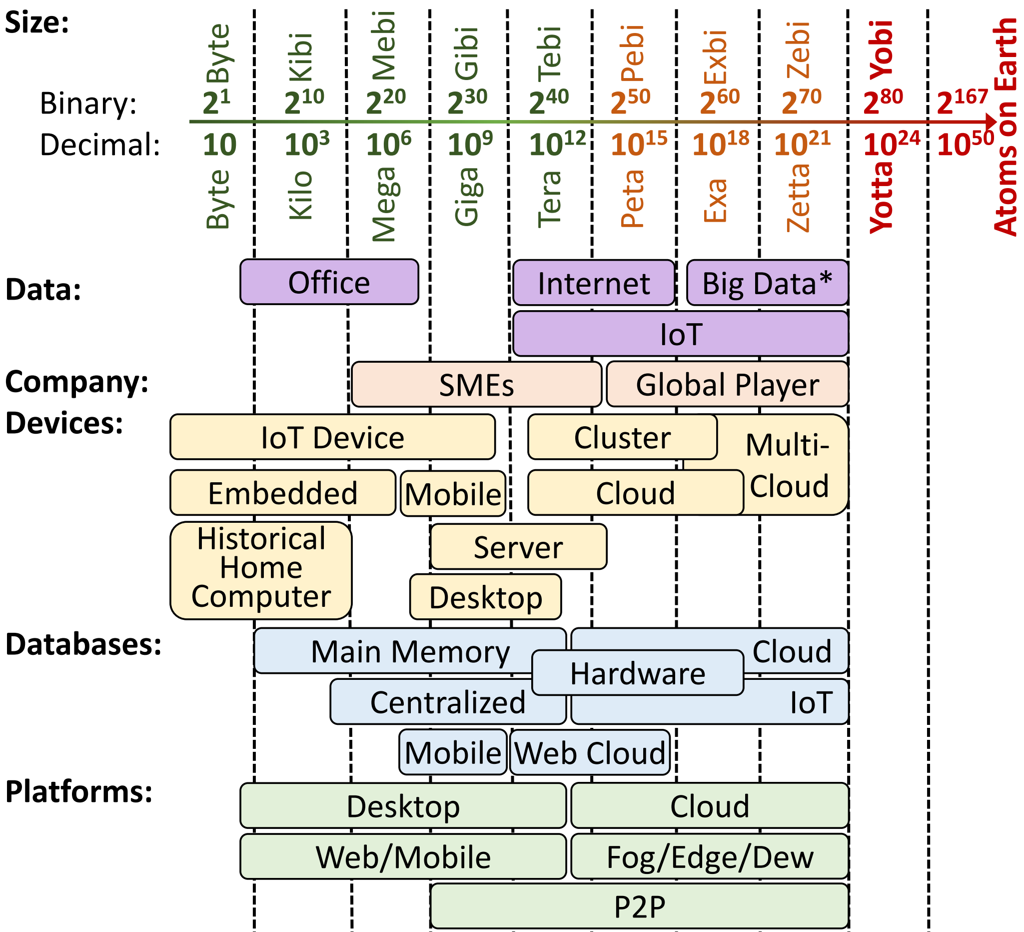
Data versus Moore





- Data sizes are growing faster than computing capacity of single CPU
  ➡ Parallel/distributed computing to overcome limitations of single CPUs

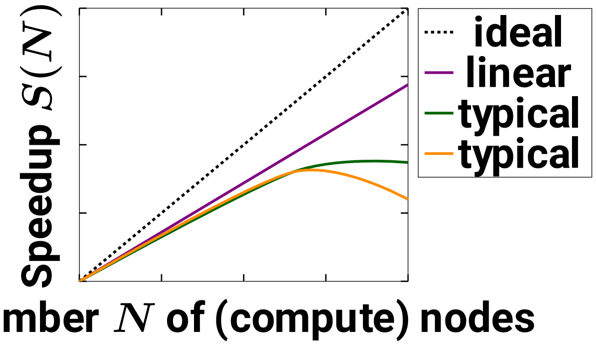
Data Sizes

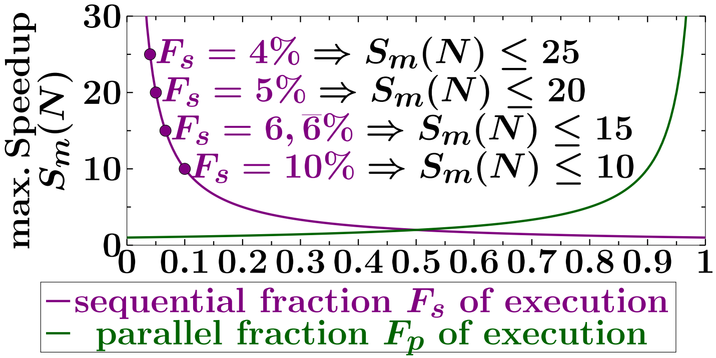


SMEs: Small and medium-sized enterprises * social media, search engines

Amdahl's versus Gustafson's law

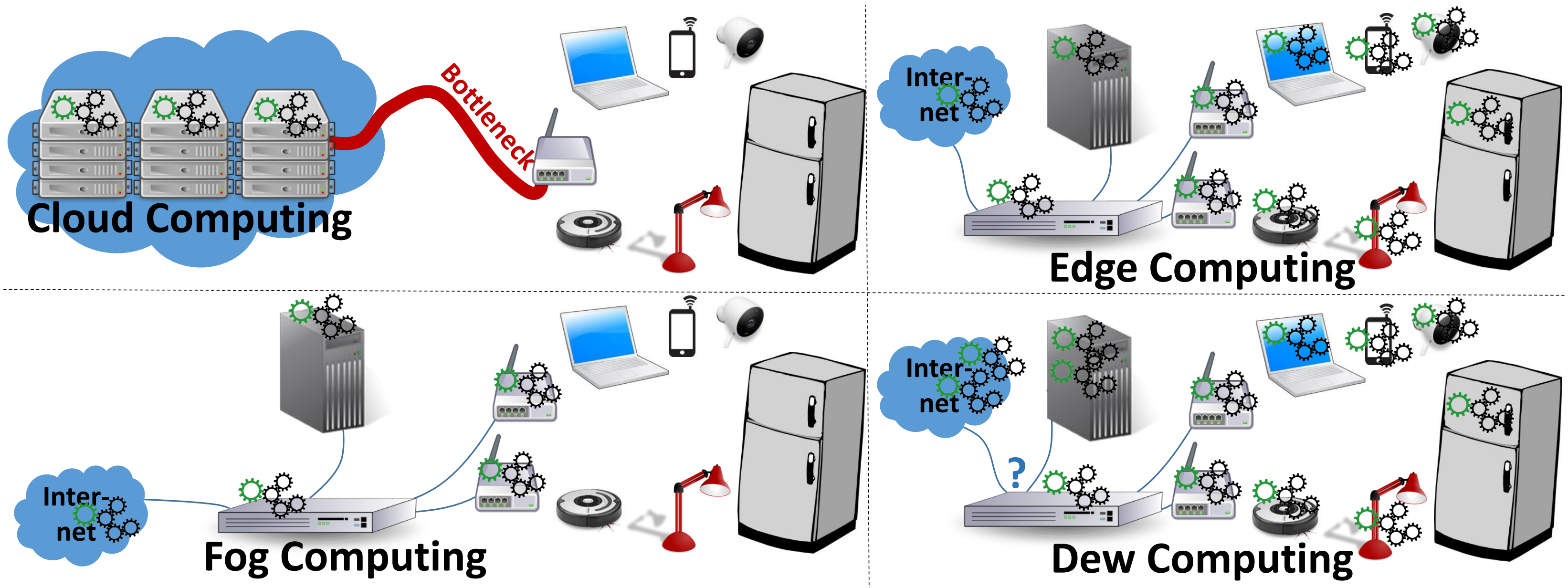
- Amdahl's law
  - a sequential part of the overall algorithm limits overall speedup (in the context of fixed problem/data size)





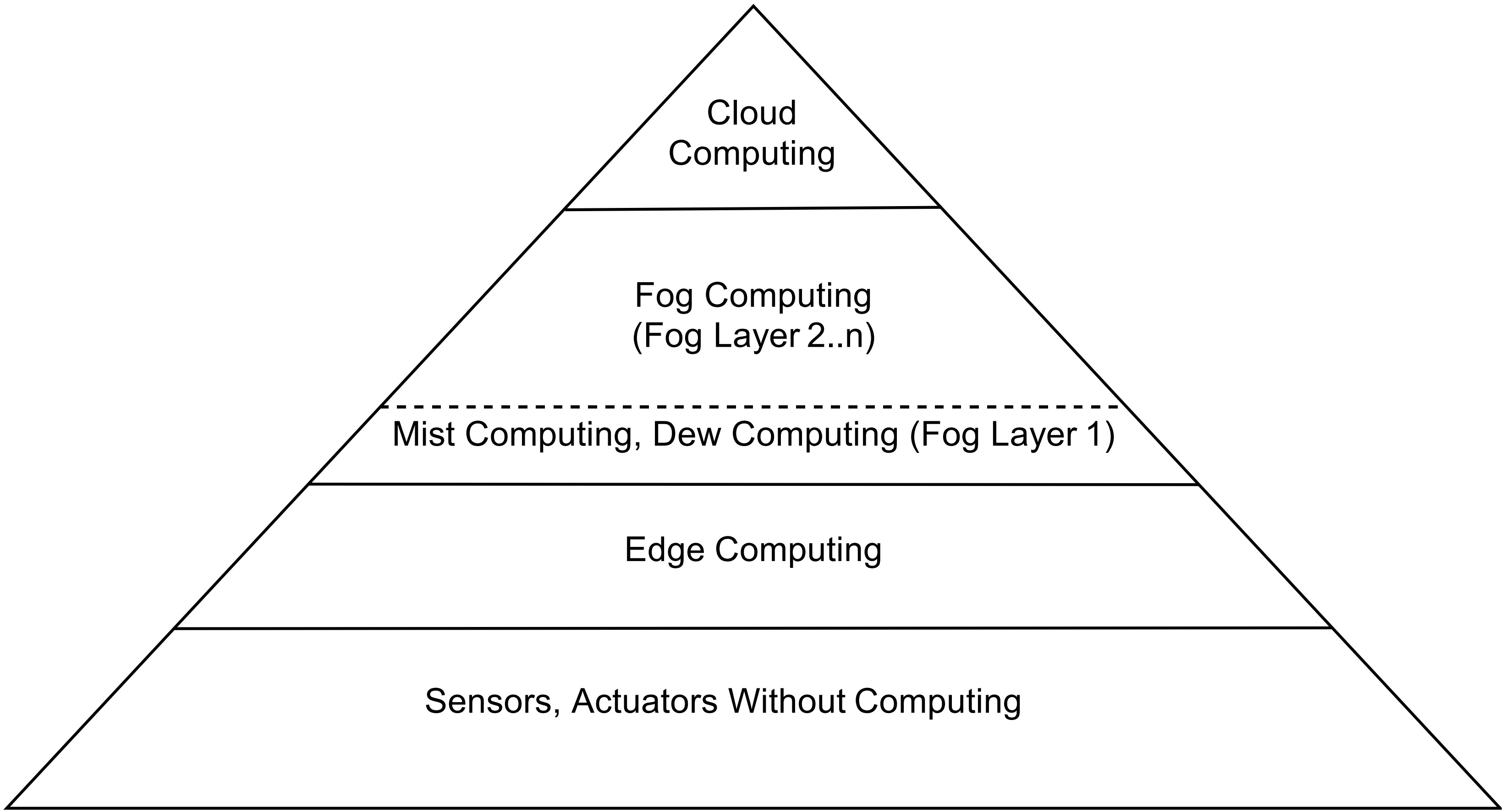
- Gustafson's law:
  - programmers tend to set the size of problems to fully exploit the computing power that becomes available as the resources improve
  - if faster equipment or more nodes are available, larger problems can be solved within the same time

Internet-of-Things (IoT) Architectures



S. Groppe: Emergent models, frameworks, and hardware technologies for Big data analytics. The Journal of Supercomputing, Vol.76(3), 2020.

Increase in #Devices from Cloud to Edge



Based on P. Varshney, Y. Simmhan, Demystifying Fog Computing: Characterizing Architectures, Applications and Abstractions. ICFEC 2017

Example of an IoT Scenario: Parking Slots

n — WSN of parking sensors that cover parking area n

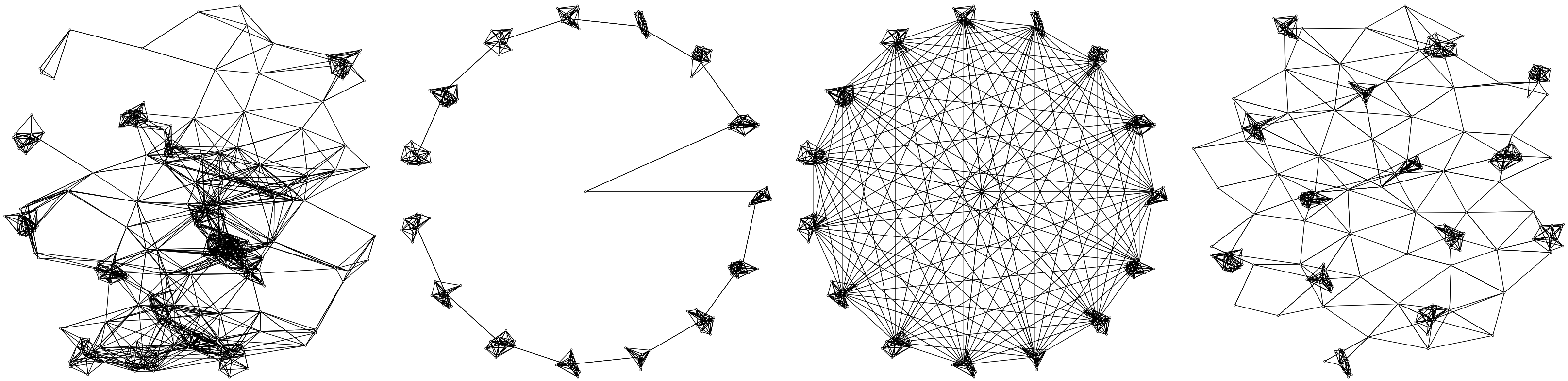
Edge-Node and gateway of the WSN

Fog-Node and gateway to the Internet

Network Topologies to be considered for IoT query processing

- Scenario: Subnets of sensors of parking slots are connected via different network topologies

Random: Ring: Full: Uniform:



Properties of Routing Algorithms

- Where is the routing performed?

Centralized	Decentralized
one or few central nodes as routers	each and every node is a router
other nodes ask the routers	nodes exchange information

- How fast do routes change?

Static	Dynamic
routes change slowly	routes change more quickly
manual configuration or restart of the routing algorithm	proactive or reactive

- How much information do the routers have?

Global	Local
router node has knowledge of the entire network	no node has knowledge of the entire network
all router nodes have the same view of the network (in a stable state)	node never knows the complete route from a source to a sink

- When is the routing performed?

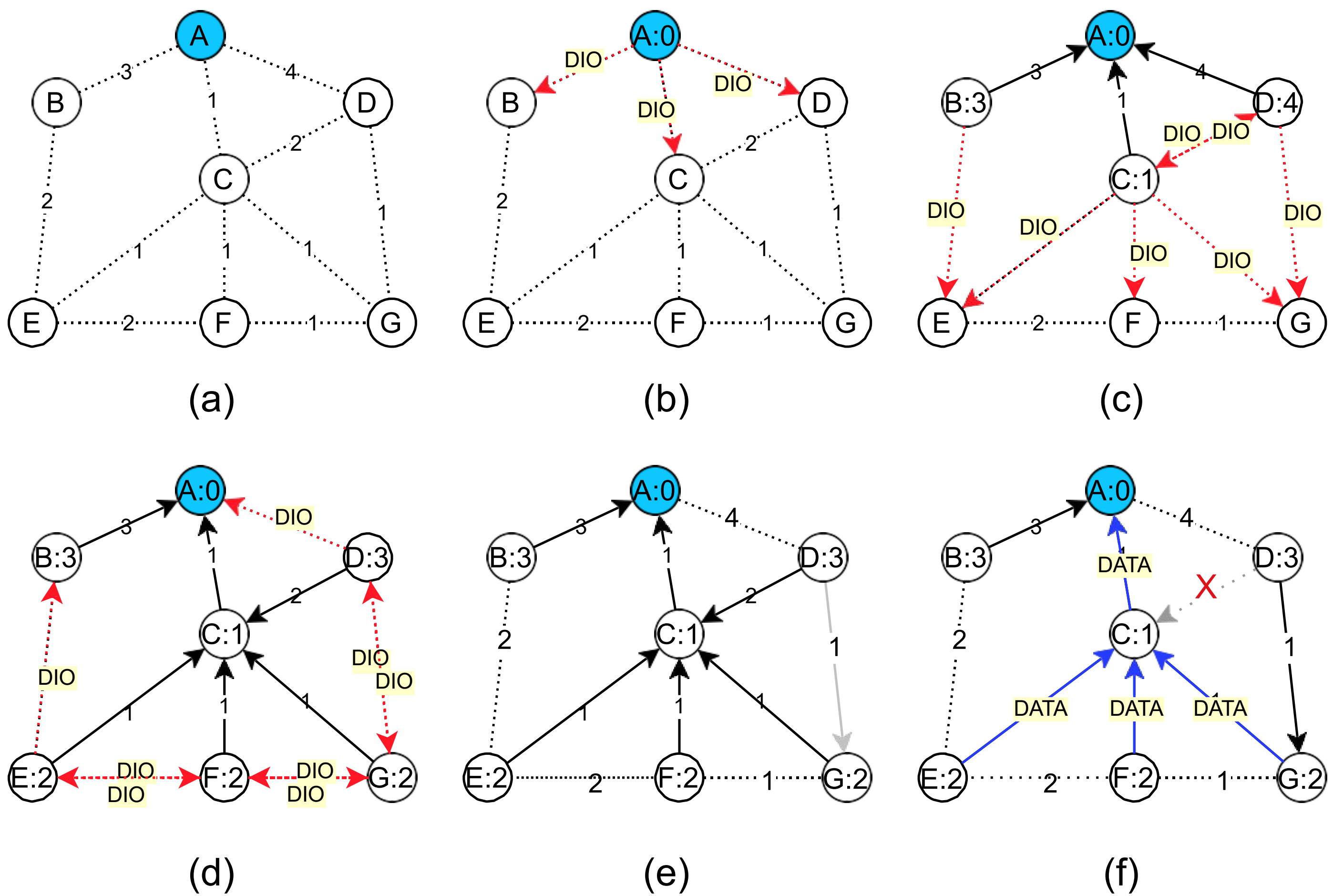
In Advance	On Demand
routing is done before forwarding	routing and forwarding is one mixed process

Routing Protocol for Low-Power and Lossy Networks (RPL)

- Construction of a Destination-Oriented Directed Acyclic Graph (DODAG)
  - via broadcast of DODAG Information Object (DIO) messages
  - similar to a sink tree in which not only the optimal routes, but also alternative routes are permitted
  - alternative routes are to protect against failures or changes in the topology and are only active when required
- Maintenance via a proactive protocol
  - Each DODAG maintains a logical clock
  - When the clock runs out, a DIO broadcast is carried out in order to track down changes in the DODAG
- Routing according to the DODAG

Sobral et al., Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications. Sensors 19(9), 2019

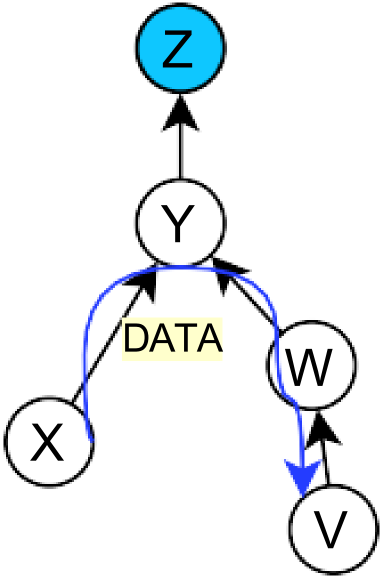
Construction of DODAG



RPL - Routing

Routing Tables and Point-to-Point Traffic in Storing Mode:

Routing Table of Z	
Destination	Next Hop
Y	Y
X	Y
W	Y
V	Y

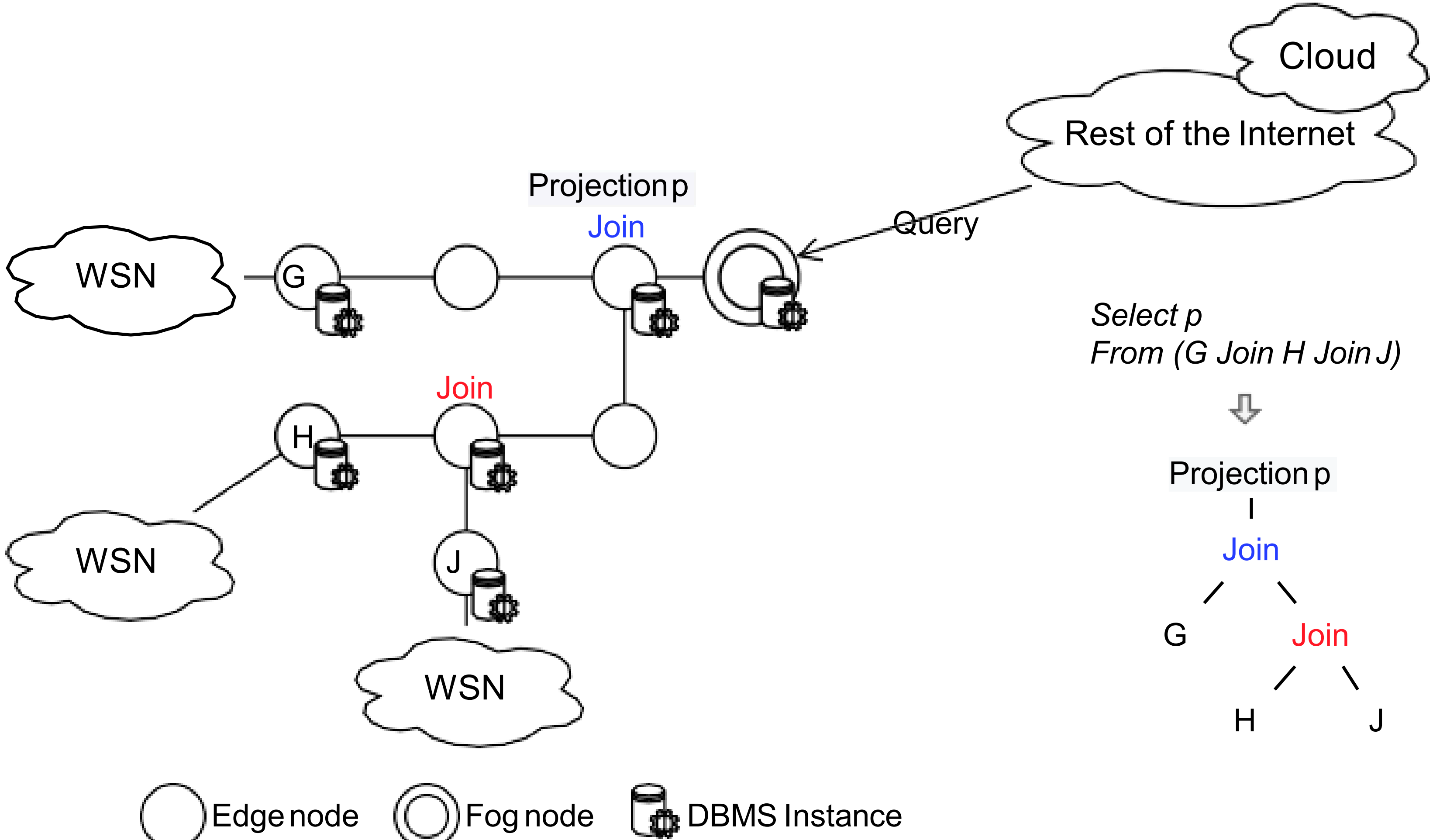


Routing Table of Y	
Destination	Next Hop
X	X
W	W
V	W

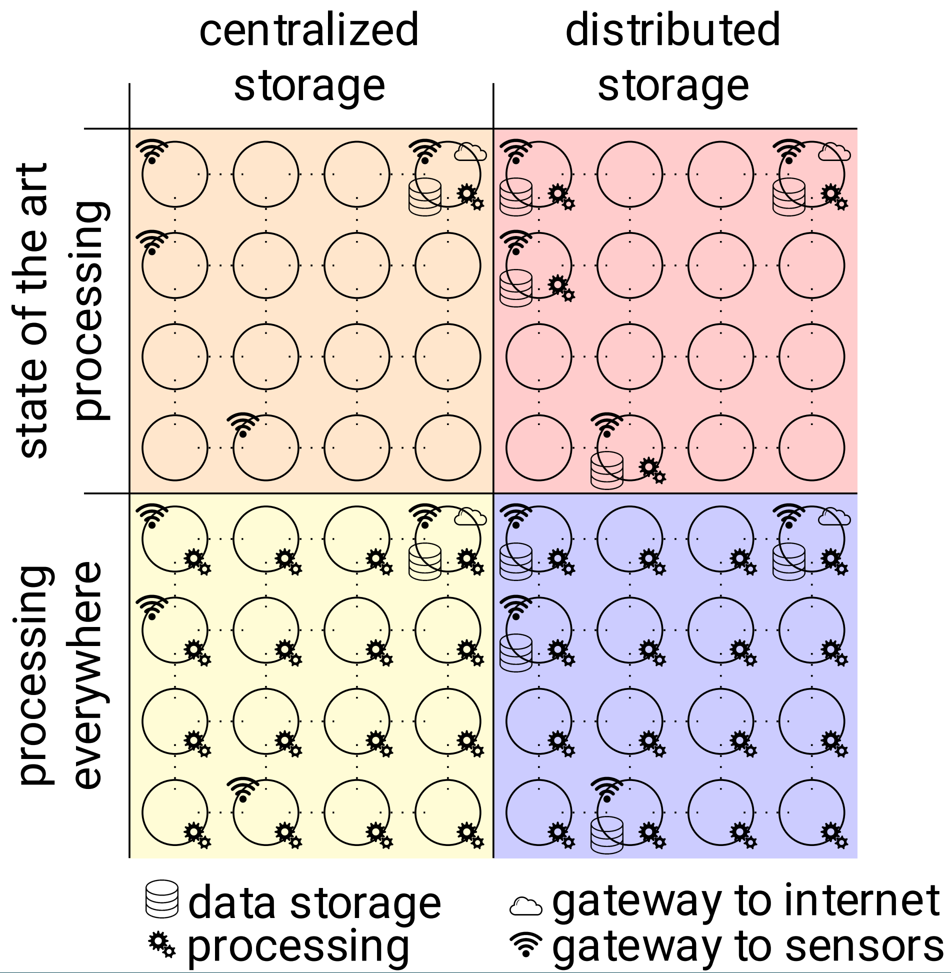
Routing Table of W	
Destination	Next Hop
V	V

Oh, S., Hwang, D., Kim, K., and Kim, K.-H. A hybrid mode to enhance the downward route performance in routing protocol for low power and lossy networks. International Journal of Distributed Sensor Networks 14(4), 2018

Embed Query Execution Plan in Topology

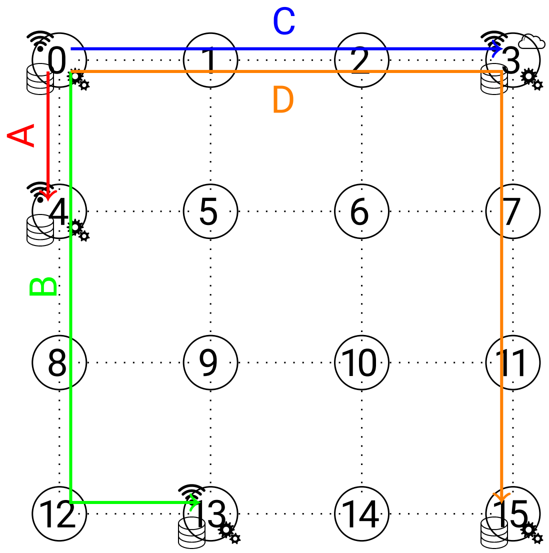


Types of Query Processing in IoT

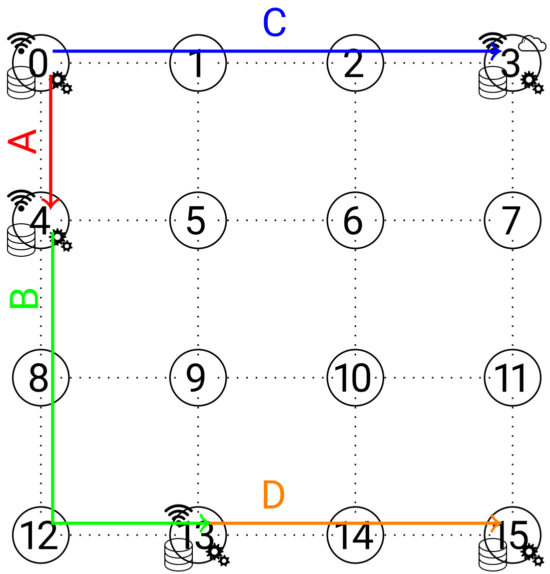


Inserting Data replicated due to indexing scheme

State-of-the-Art:



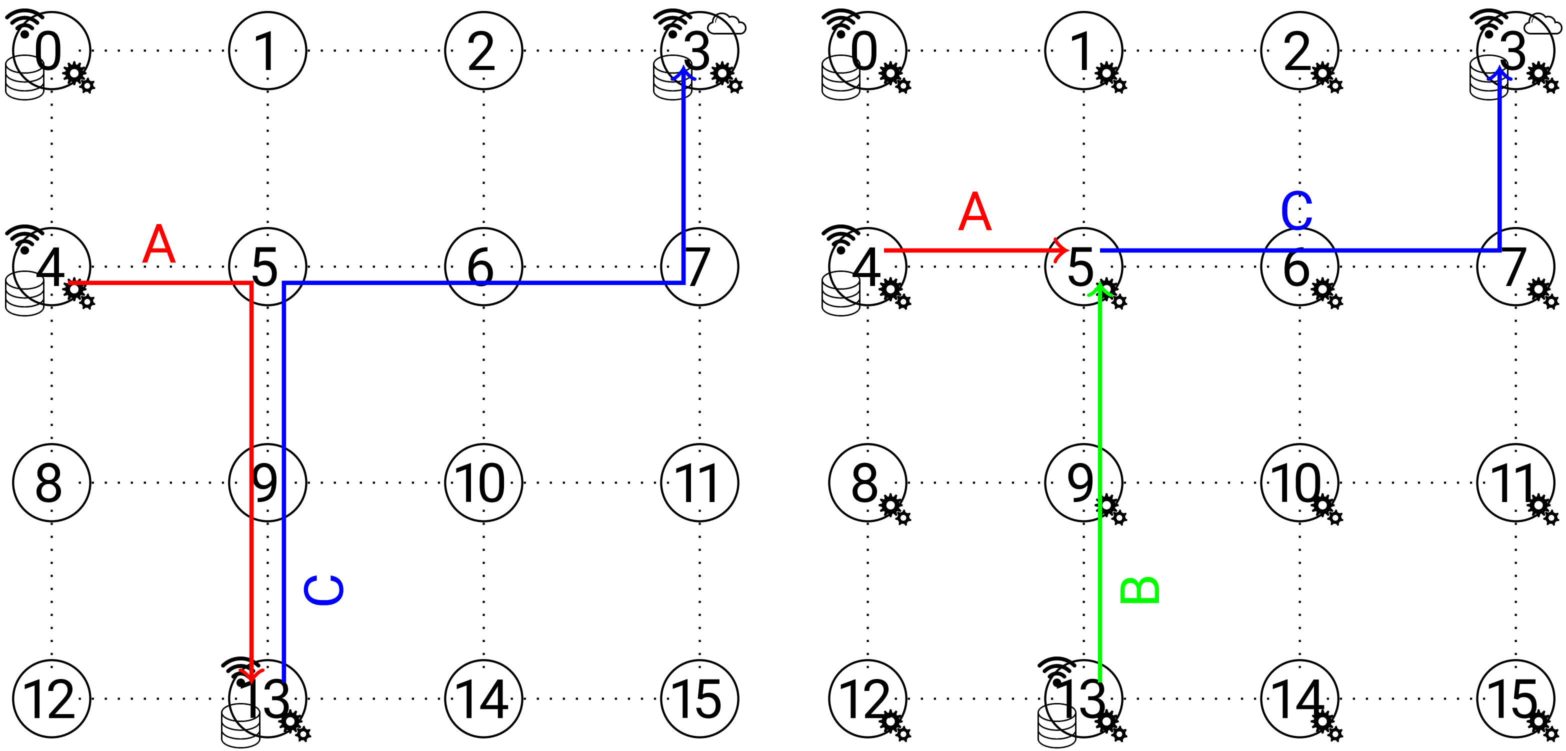
Multi-Cast:



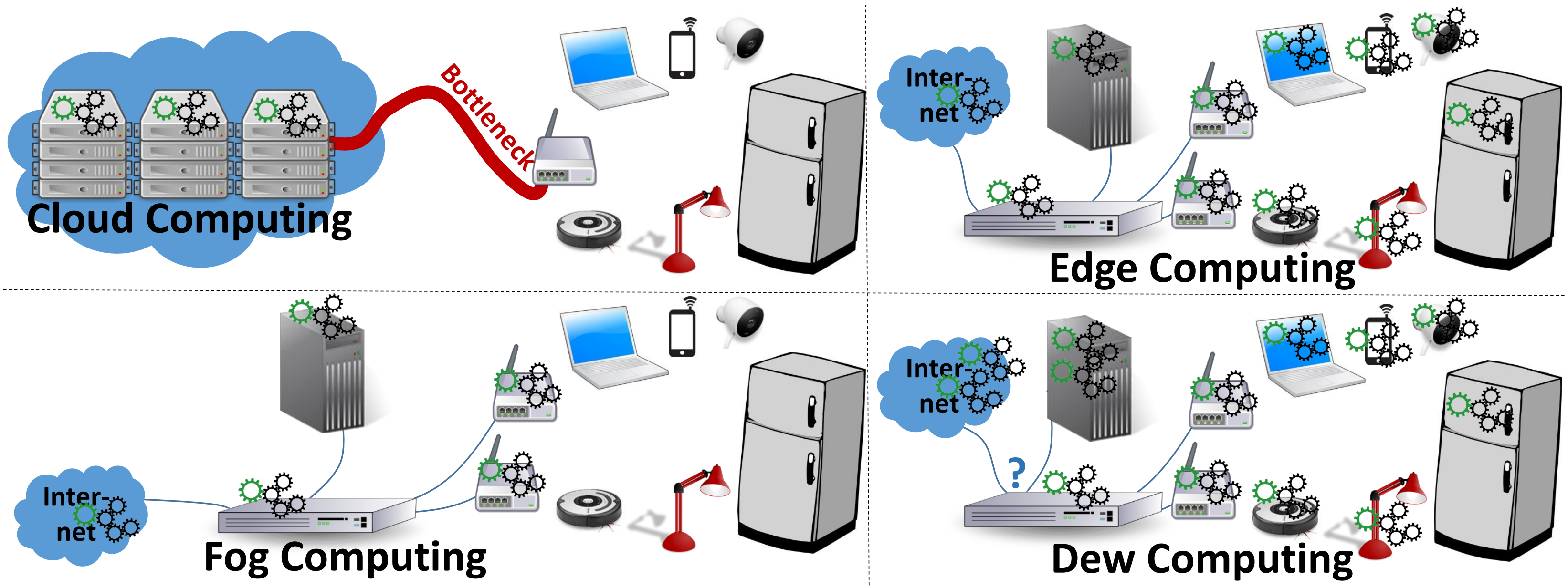
- network traffic reduction during insert by using
  - IPv6 Routing Protocol for Low-Power and Lossy Networks (**RPL**)
    $\longrightarrow$ All-Shortest-Paths (**ASP**): **17-48%**
  - **multi-cast**: **24%**
  - **additional devices**: **3%**

Querying Data in IoT $(4 \bowtie 13 \rightarrow 3)$

State-of-the-Art: Combining Routing & Processing:

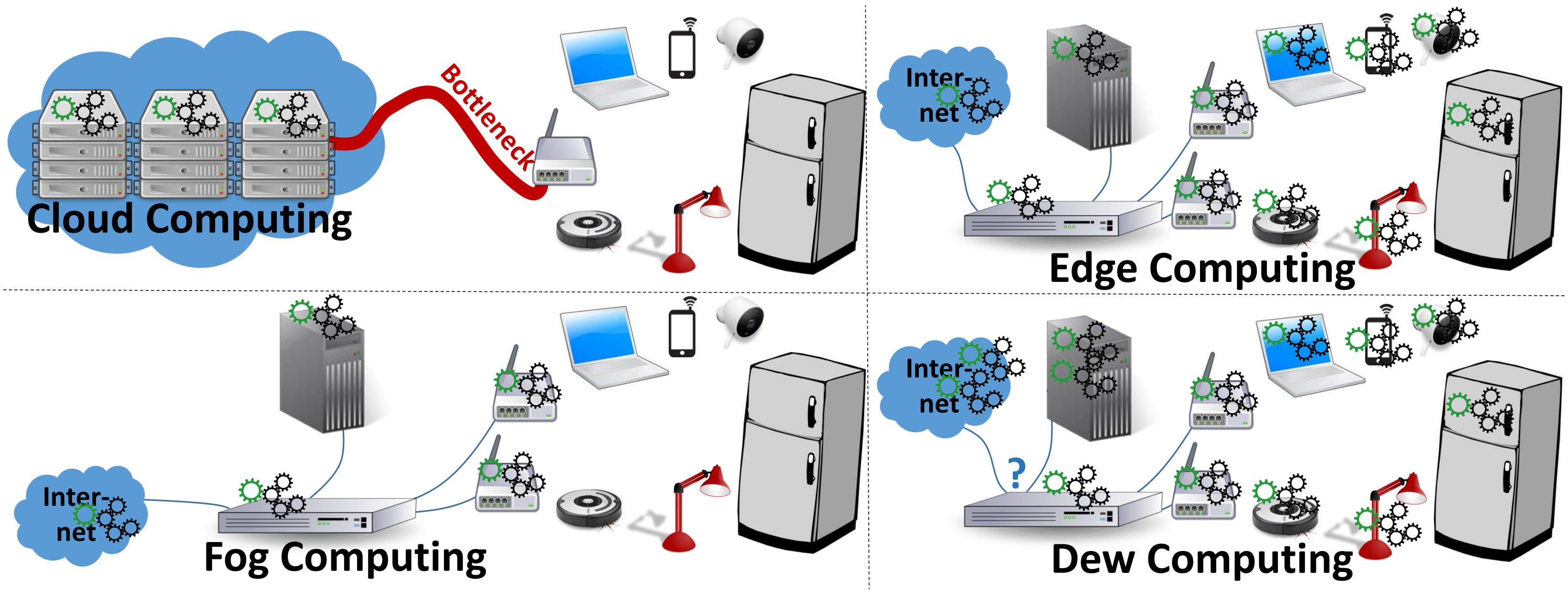


IoT and Cloud as Platforms for Processing the IoT DATA



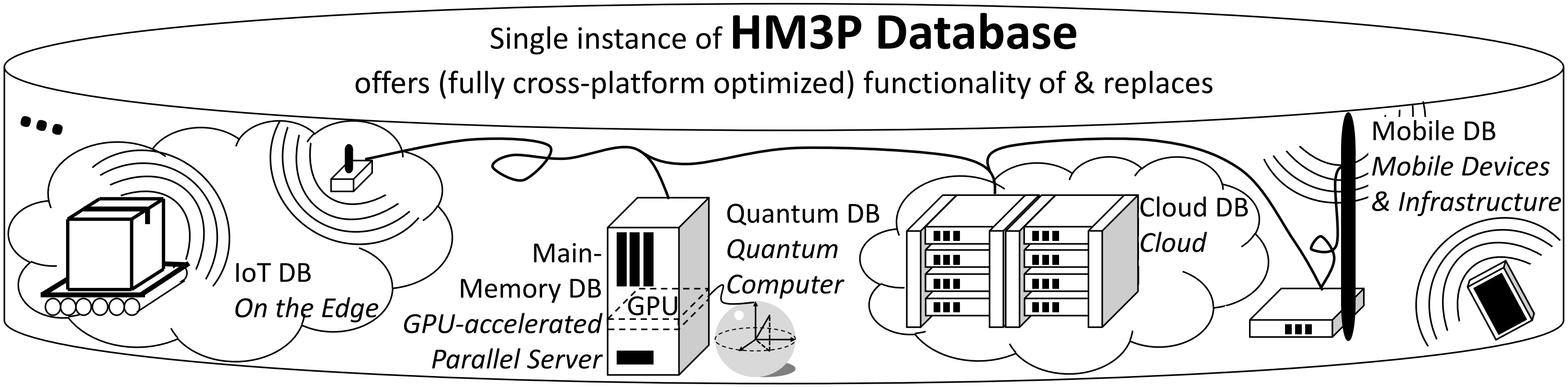
S. Groppe: Emergent models, frameworks, and hardware technologies for Big data analytics. The Journal of Supercomputing, Vol.76(3), 2020.

IoT and Cloud as Platforms for Processing the IoT DATA



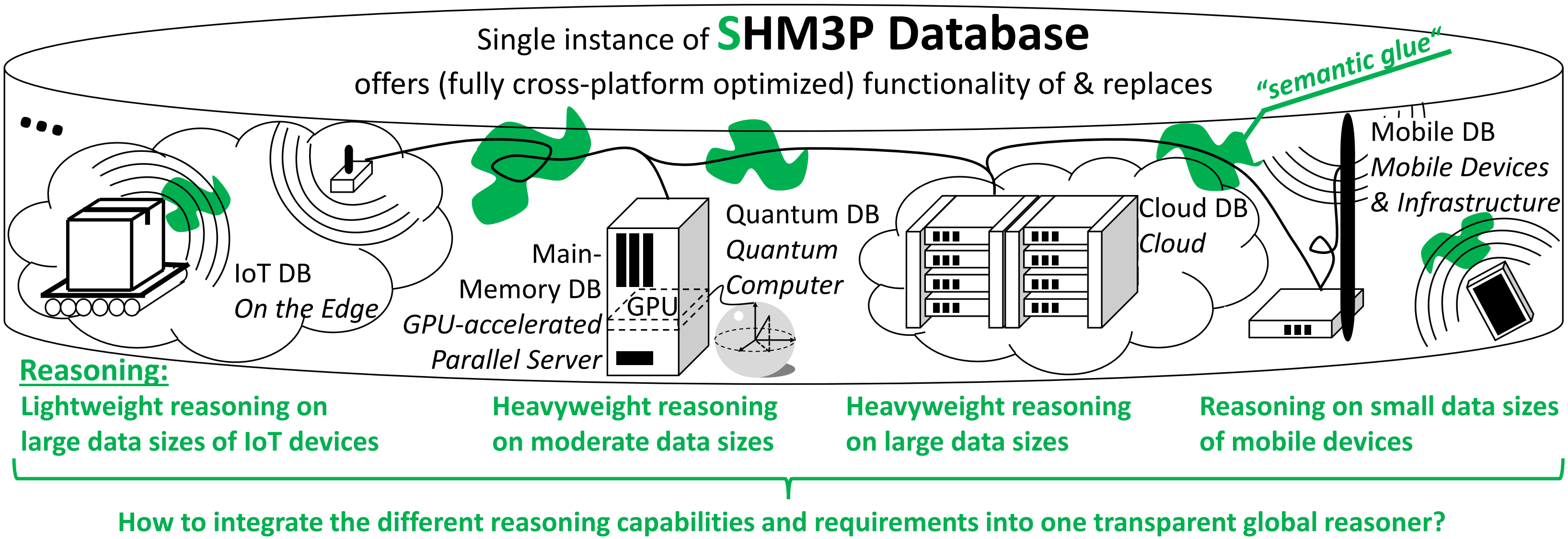
- ...but there are also **other platforms**!

Hybrid Multi-Model Multi-Platform (HM3P) Database



- full and uniform data integration at database level
- performance: fully optimized across different data models
- transparent fault-tolerance
- SQL standards: relational ('87), XML ('03), temporal ('11), JSON ('16), Multi-dimensional Arrays ('19), schemaless ('19), streams ('20?), property graphs ('21?)
- **features of different types of databases running on different platforms can be used**

Variant: Semantic HM3P (SHM3P) DB



- Semantic Layer as glue between other models and platforms
- new challenges like integrating different types of reasoners in a transparent global reasoner

+ Features of HM3P databases
+ Easier data integration
- Performance issues may occur due to semantic layer

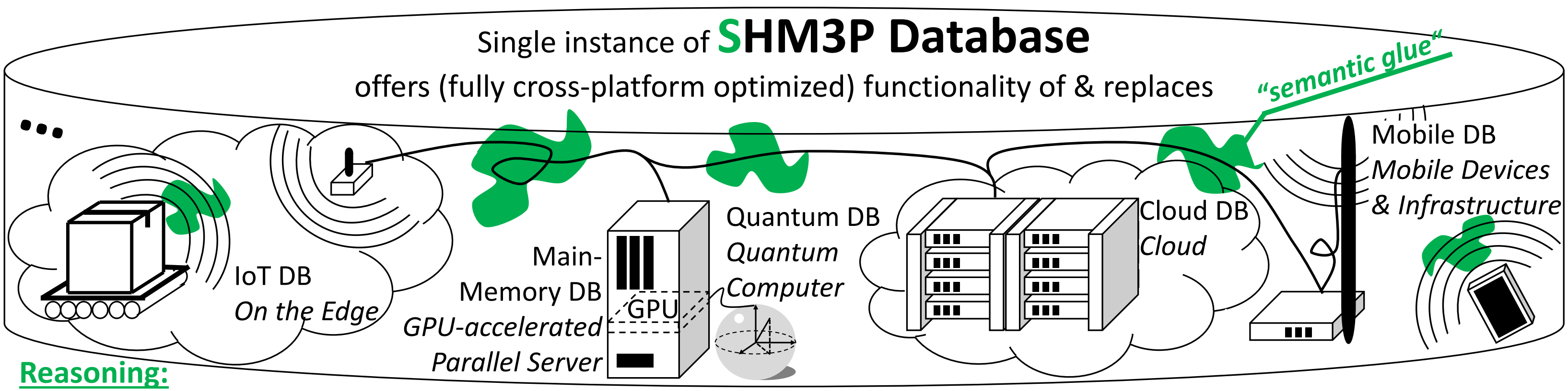
Challenges for HM3P Databases 1/2

- developing only one code base for the different platforms, but not introducing performance overhead in comparison to single platform databases
- identifying common properties of several platforms and reusing those approaches (like fault tolerance mechanisms) in different combinations, which are best suitable for these considered platforms
- data distribution among different platforms (applying different data distribution approaches as well)
- data distribution strategies considering overall the different properties of used platforms and models (like fast reads on parallel servers (using relational databases) and fast updates in cloud databases)

Challenges for HM3P Databases 2/2

- query optimization and other database tasks across different platforms, which apply different database approaches
- dealing with and integrating different privacy and security mechanisms supporting different privacy and security levels in the different platforms (with research e.g. on querying heterogeneous encrypted data)
- concurrency control approaches of different type have to be combined and work in cooperation (like 2 phase locking for server platforms and optimistic concurrency control for P2P networks)
- combining different types of databases (on different platforms) to offer the best of these databases and platforms *under one hood* to applications and users transparently or via intelligent integration into query language and API, e.g.,
  - guaranteeing atomicity and isolation in transactions for the data stored on a parallel server, but not for those data in the cloud supporting fast updates

Semantic Hybrid Multi-Model Multi-Platform (SHM3P) Database



Reasoning:

Lightweight reasoning on large data sizes of IoT devices

Heavyweight reasoning on moderate data sizes

Heavyweight reasoning on large data sizes

Reasoning on small data sizes of mobile devices

How to integrate the different reasoning capabilities and requirements into one transparent global reasoner?

- How to integrate the semantic layer between different types of databases and support semantic processing specialities like reasoning over the boundaries of different platforms?

Challenges for SHM3P Databases

- integrating different data models in a semantic layer on top of the underlying data models
- efficient transformations from and to the semantic model in an operational system
- developing efficient semantic querying and reasoning over the integrated data of different models
- global reasoning over reasoners running on different platforms supporting some kind of distributed heterogeneous reasoning
- developing a combination of stream reasoning over streaming data (e.g. of IoT devices) with static reasoning over large-scale data sets (stored e.g. in clouds)
- supporting transactions over semantic data by integrating the reasoner in transaction synchronization

Summary and Conclusions

- Internet-of-Things as extension of the Internet for advanced applications by connected things
- Routing Protocols with RPL as example
- Query Processing as example for data management tasks
- Combining routing and query processing for reducing communication costs as example for new research in IoT by combining approaches of the network and database community
- The bigger picture: Combining approaches designed for various platforms with specific properties and benefits