Bachelor-/Master-Forum 2021

Utilizing Emergent Technologies for Big Data Analytics

Institut für Informationssysteme (IFIS)

15.12.2021

Professor Dr. rer. nat. habil. Sven Groppe
https://www.ifis.uni-luebeck.de/index.php?id=groppe

Institut für Informationssysteme (IFIS)

- **Professoren**
  - **Prof. Dr. rer. nat. Sven Groppe**
  - **Prof. Dr. rer. nat. Ralf Möller** (Direktor)
- Wissenschaftliche/Projekt-Mitarbeiter*innen
  - Magnus Bender, M.Sc.
  - Marcel Gehrke, M.Sc.
  - Dr. rer. nat. Jinghua Groppe
  - Tobias Groth, M.Sc.
  - Mena Leemhuis, M.Sc.
  - Malte Luttermann, M.Sc.
  - Dr. rer. nat. Sylvia Melzer
  - **PD Dr. rer. nat. habil. Özgür Özçep**
  - Simon Schiff, M.Sc.
  - Benjamin Warnke, M.Sc.
- Weitere Doktorand*innen
  - Fabian Bormann, M.Sc.
  - Nils Finke, M.Sc.
  - Mattis Hartwig, M.Sc.
  - Marisa Mohr, M.Sc.
  - Simon Paasche, M.Sc.
  - Maurice Sambale, M.Sc.
  - Mareike Stender, M.Sc.

in bold: **formal supervision** of bachelor/master thesis, URL: **www.ifis.uni-luebeck.de**

Contacts for Bachelor/Master Thesis

- For topics please check:
  - https://www.ifis.uni-luebeck.de/index.php?id=studentische-arbeiten
- Prof. Ralf Möller (his topics are not in this talk, please contact him)
  - Dynamic Probabilistic Relational Models
  - Probabilistic Computing
  - other topics (artificial intelligence/machine learning)

- PD Özgür L. Özçep
  - Logical and mathematical Modeling for Knowledge Representation and Processing and for Machine Learning
  - Ontology-based Data Access (OBDA)

- Prof. Sven Groppe
  - Utilizing Emergent Technologies (e.g., GPU, Quantum Computing) for Big Data Analytics
  - Databases
  - Semantic Web
  - IoT Data Management

Zoo of Data Formats, for example:

- relational data
  - in relational databases
- XML
  - for exchange
- JSON
  - web data
- Resource Descr. Framework (RDF)
  - Semantic Web
- graph data
  - from social networks
- unstructured data
  - of social media like wikis

➡ Parallel use of different Data Models for storing and processing

Relational:

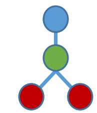
Primary		Secondary		Primary	

XML:

```
<root>
 <child>
  <first>hello</first>
  <sibling>sibling</sibling>
 <child>
</root>
```

JSON:

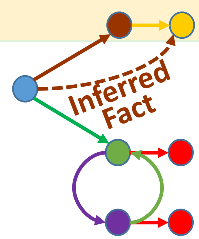
```
{root:{
    child:{
      first:hello,
      sibling:sibling
    }
}}
```



RDF/Graph Data:

Ontology of Semantic Web

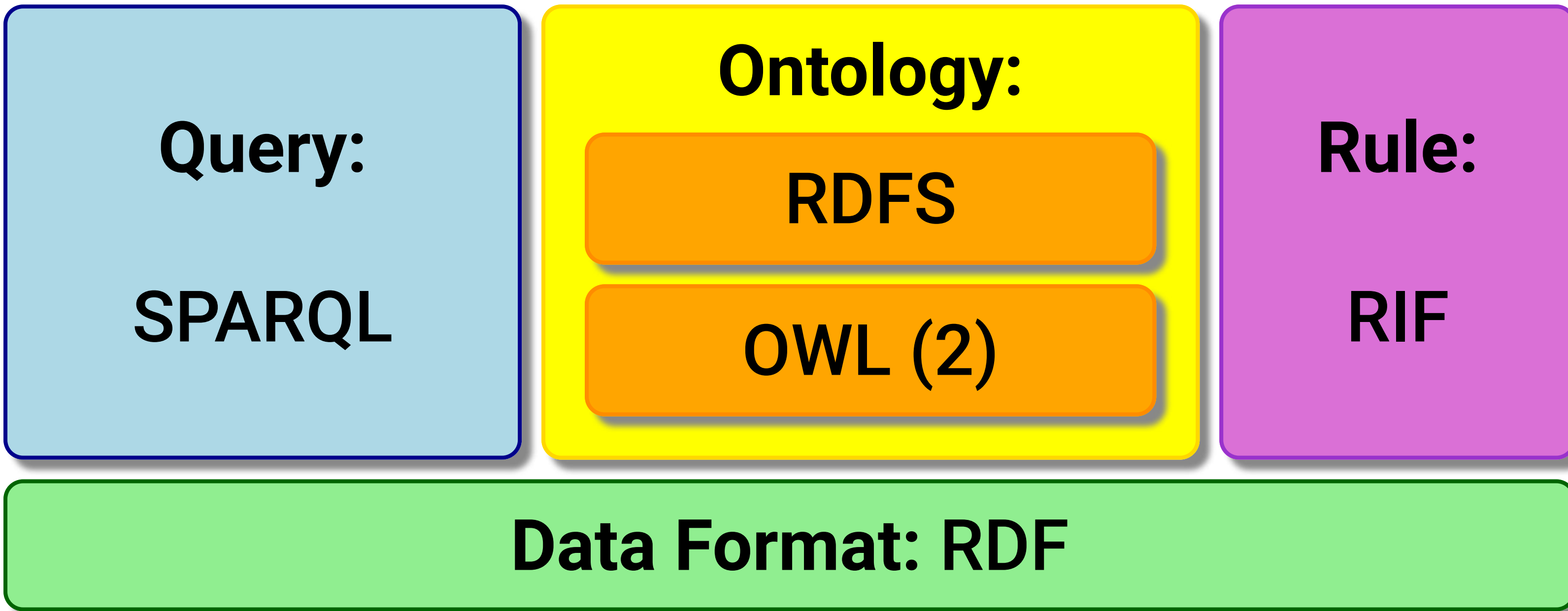
```
:article   rdfs:subclassOf bench:doc
:article1  rdf:type        :article .
:article1  dc:creator      :person1 .
:person1   foaf:name       'Martin' .
:person1   :likes          :person2 .
:person2   foaf:name       'Jennifer' .
:person2   :hates          :person1 .
```



Unstructured Data:

Title
The following issues are important:
1. Very Importand Persons (VIPs)
2. Very Important Data (VID)

Semantic Web (Core) "Standards"



- Every data model (here Semantic Web) has its own set of languages (data, query, rule, ...)

Semantic Web: Ontology

- Ontology as additional abstraction layer
  - More than schema descriptions:
    - Specification of background knowledge (based on which new facts can be derived)

      $\Rightarrow$ avoids storing of redundant data

      $\Rightarrow$ supports re-use of data

      $\Rightarrow$ supports data integration

      $\Rightarrow$ **increases computational complexity**

Semantic Web: Ontology

- Ontology as additional abstraction layer
  - More than schema descriptions:
    - Specification of background knowledge (based on which new facts can be derived)

      $\Rightarrow$ avoids storing of redundant data

      $\Rightarrow$ supports re-use of data

      $\Rightarrow$ supports data integration

      $\Rightarrow$ **increases computational complexity**

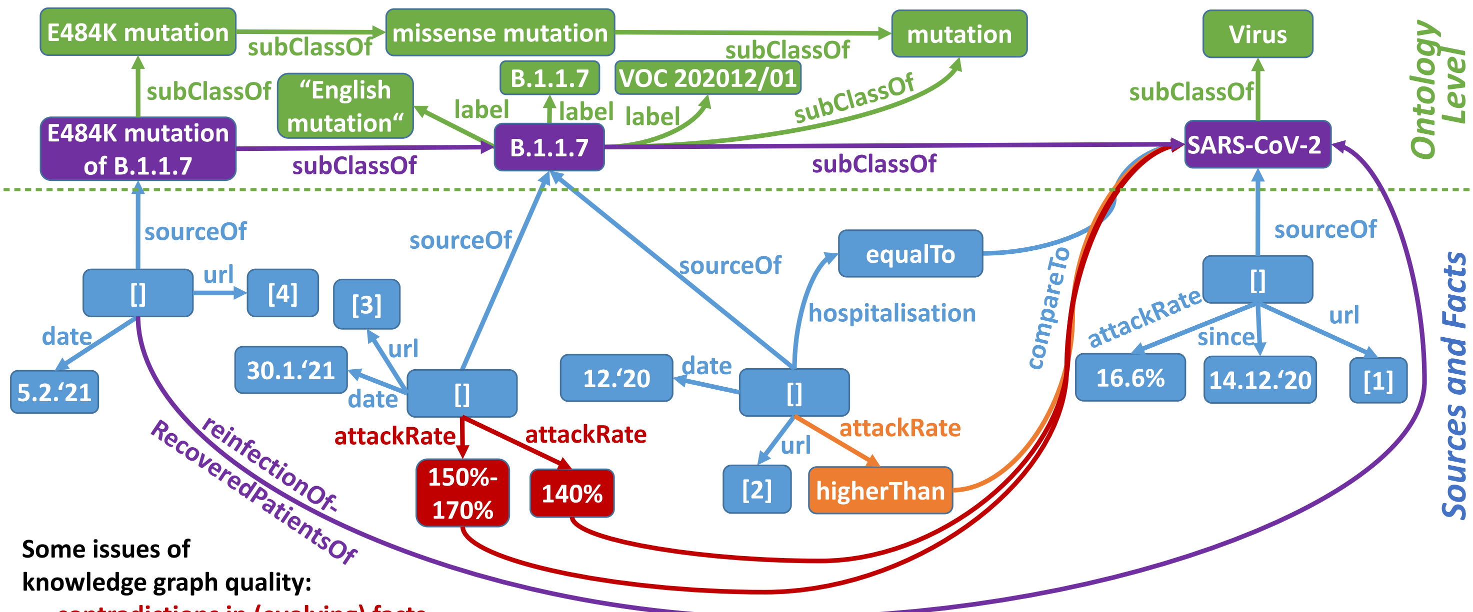
- However, ontologies do not pose strict restrictions on RDF graphs
  - SHACL Shapes Constraint Language: for validating RDF graphs against a set of conditions

St. Arbeiten durch PD Özçep betreut

Wissenschaftl. Mitarbeiterin: Mena Leemhuis

- angesiedelt im Rahmen des BMBF-Verbundprojekts **"Digitalisierung smarter Materialien und ihrer Herstellprozesse (SMADI)"**
- Ontologie-basierter Zugriff auf Daten und Modelle (OBDMA)
  - Validierung von SHACL-Integritätsbedingungen
  - Mapping-Generierung auf der Basis von SHACL-Integritätsbedingungen
  - Optimierung des OBDMA-Transformationsalgorithmus mit SHACL-Integritätsbedingungen
  - Optimierung des OBDMA-Transformationsalgorithmus mit dem Magic-Set-Verfahren
  - Rollenkomposition
  - Anfrageupdate
  - Bedingte Anfragen

Thesis supervised by Prof. Groppe
COVID-19 Knowledge Graph (KG)



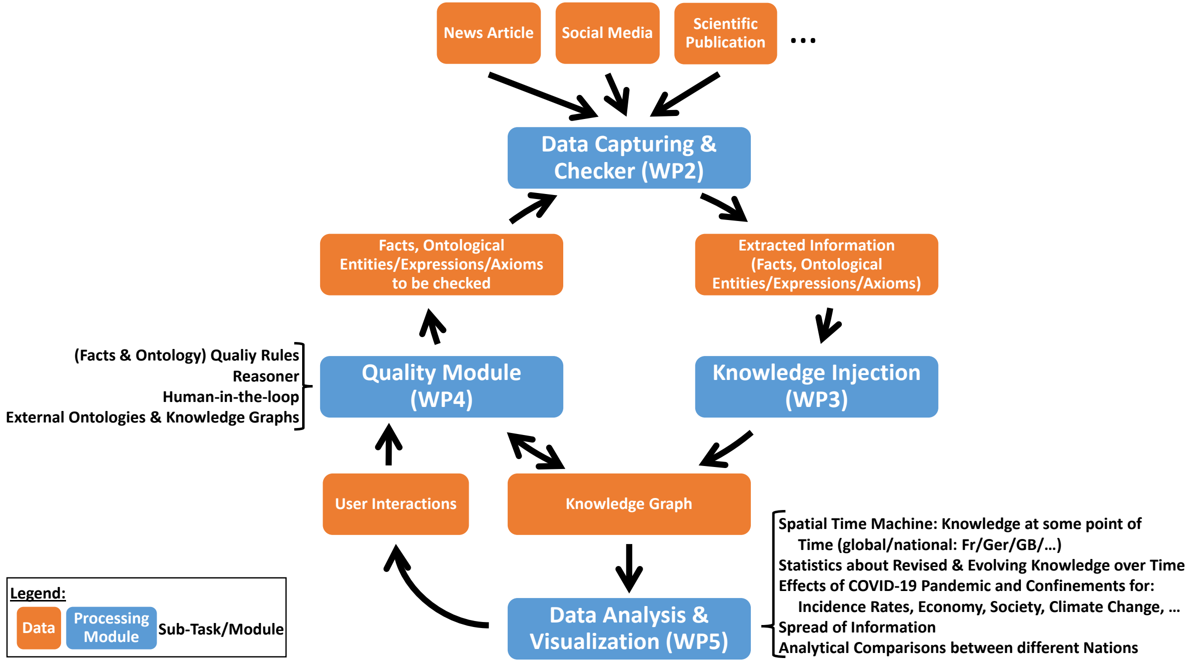
Some issues of knowledge graph quality:

- contradictions in (evolving) facts
- checking vague formulations and compare them with other given information
- errors with ambiguity: E484K mutation of B.1.1.7 is a SARS-CoV-2 virus, but reinfection with E484K is only possible for patients recovered from SARS-CoV-2 virus not mutating E484K (→ introduce class "SARS-CoV-2 without E484K mutation")

[1] https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2774102
[2] https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/959361/Technical_Briefing_VOC202012-2_Briefing_2.pdf
[3] https://www.ruhr24.de/service/corona-britische-mutation-neue-studie-mutante-virus-toedlicher-sterberate-deutschland-90184403.html
[4] https://www.bmj.com/content/372/bmj.n359

High Quality Knowledge Graphs from recent English, French and German Emergent Trends with the example of COVID-19

- Project (DFG/ANR) starting soon (partners: universities of Paris, Toulouse and Lübeck)
- Thesis in the areas of
  - data capturing,
  - visualization & analysis,
  - detection of contradictions in KG, ...

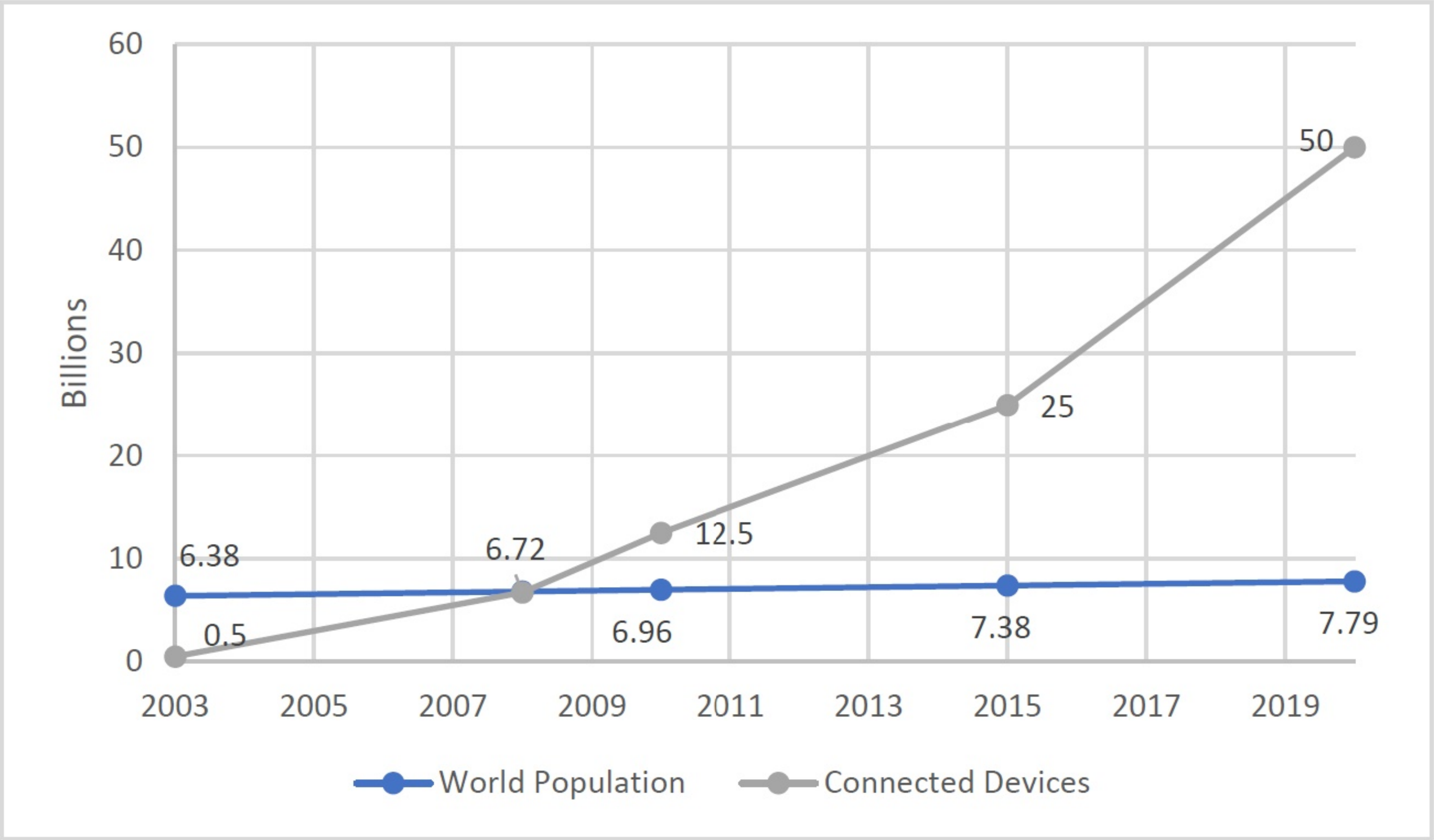


Definitions of Internet-of-Things (IoT)

- 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



Data Sources: **#Connected Devices** **#Humans on Earth**

Things in IoT are heterogeneous

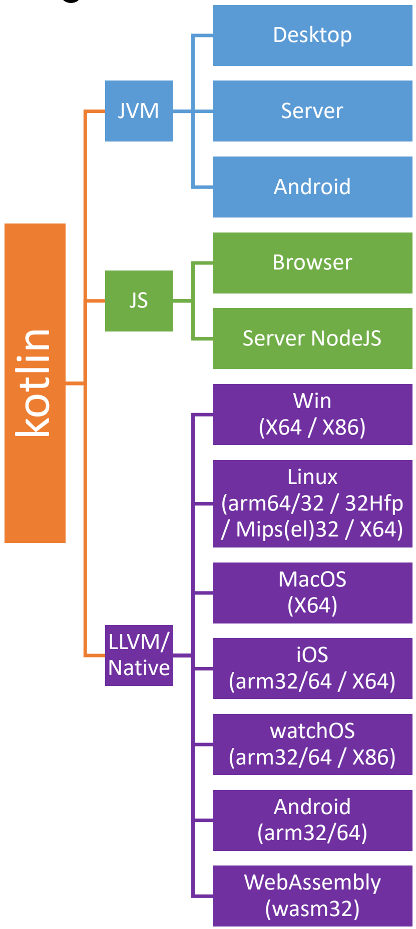
- Different kinds of devices: Smartphone/-watch-/tv/..., sensors, routers, tablets, raspberry pi (and similar), ...
- Hardware (e.g. CPU)
- Operating system (OS)
- Network protocols
- Capabilities
  - Performance of CPU
  - Storage capacity
  - Access to networks (wireless, stationary, ...)
  - ...
-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

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

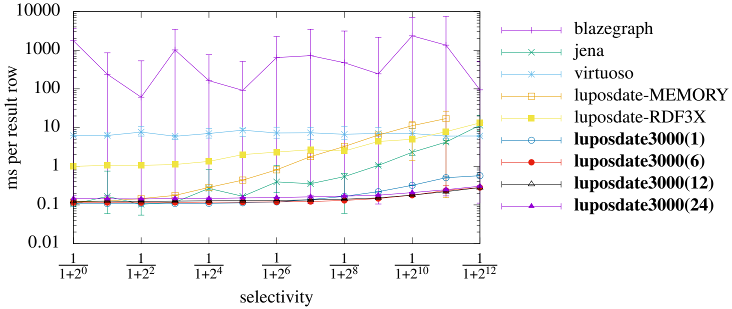
Targets:



- 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)

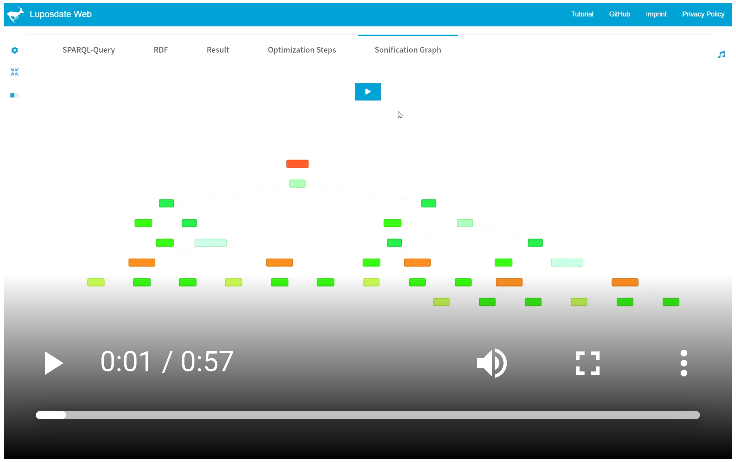
The Power of Multi-Platform: LUPOSDATE3000

- ultra-fast in jvm...



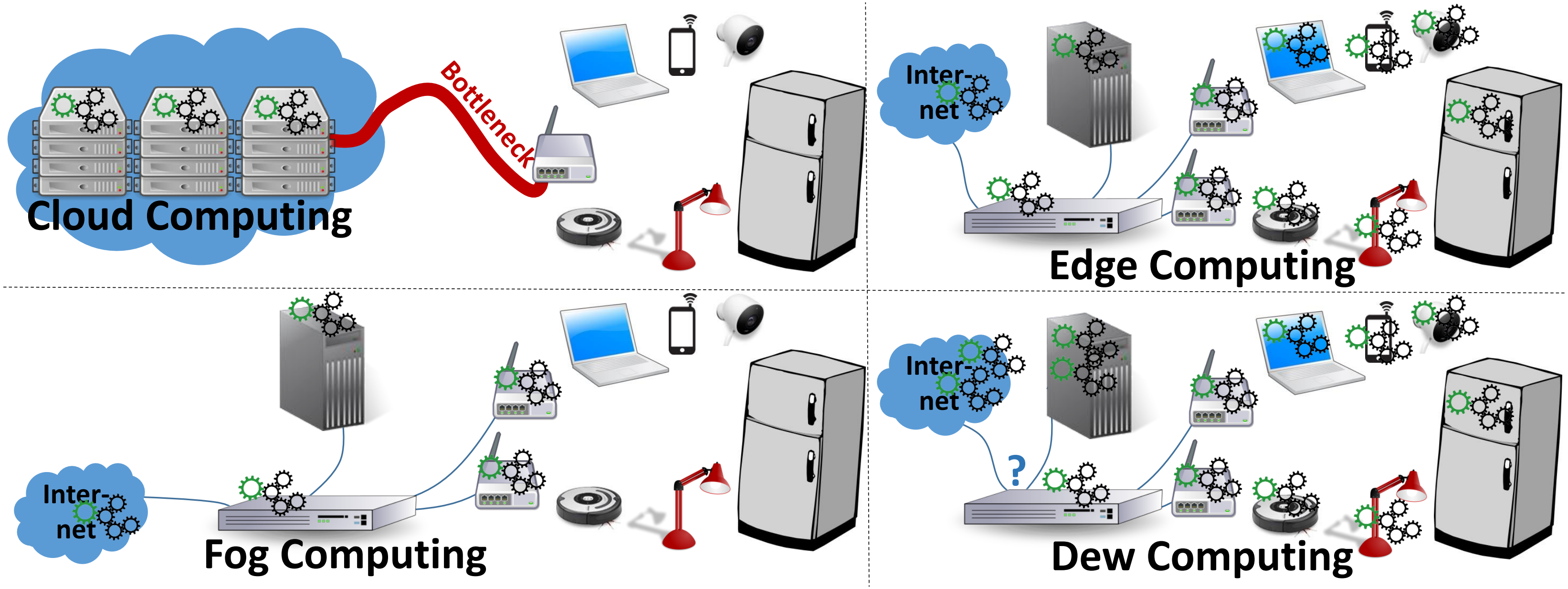
B. Warnke, M.W. Rehan, S. Fischer, S. Groppe: Flexible data partitioning schemes for parallel merge joins in semantic web queries in: BTW'21

- ...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

Internet-of-Things (IoT) Architectures

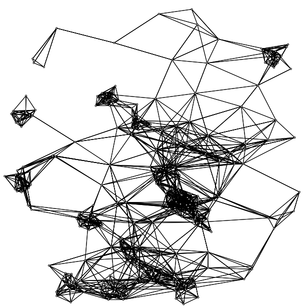


Example of an IoT Scenario: Parking Slots

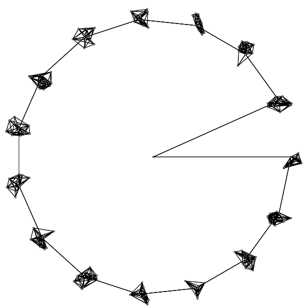
Network Topologies to be considered for IoT query processing

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

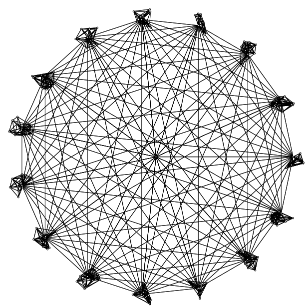
Random:



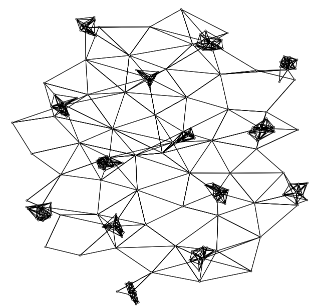
Ring:



Full:



Uniform:



Warnke et al., SIMORA: SIMulating Open Routing protocols for Application interoperability on Edge Devices, submitted

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

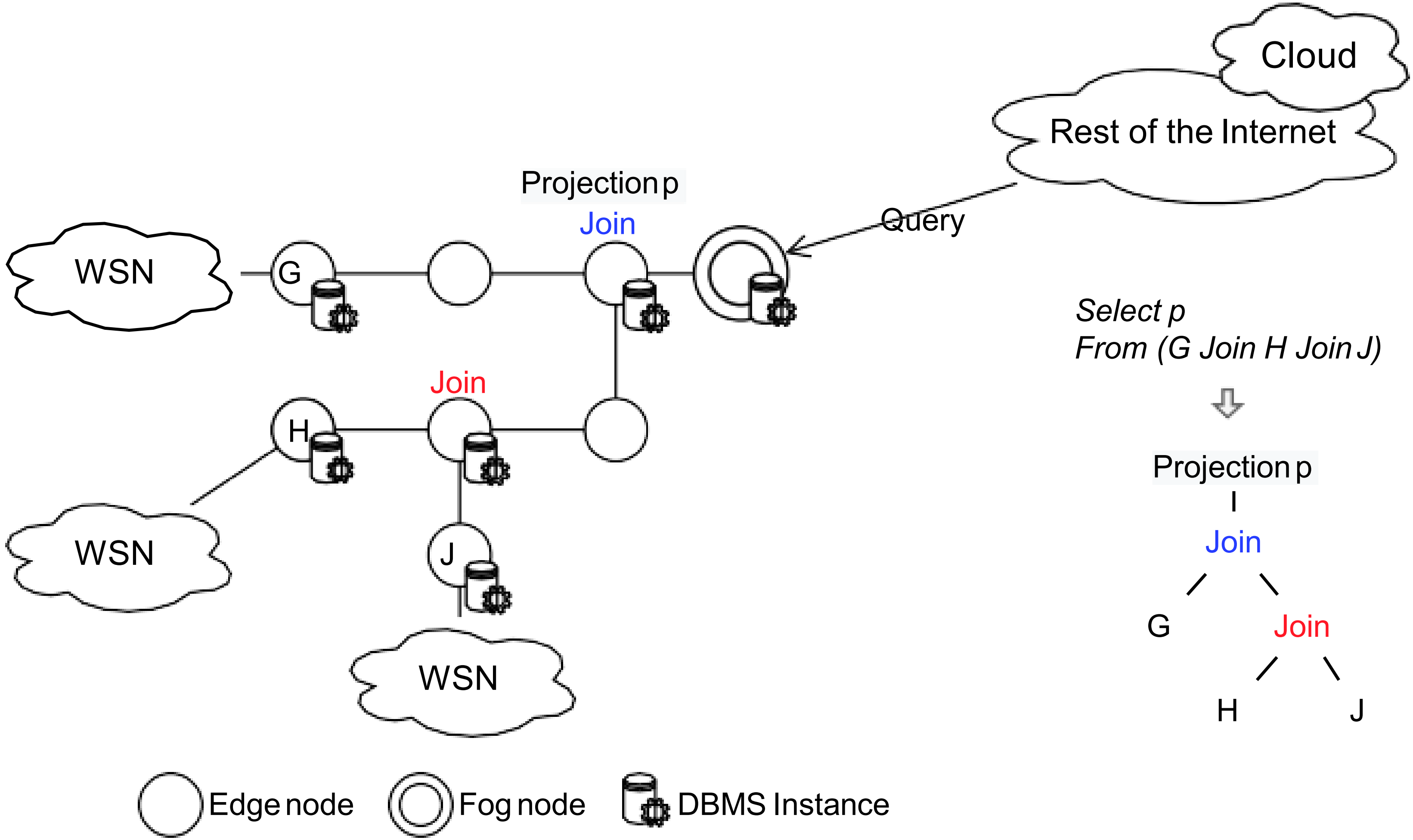
- How much information do the routers have?

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?

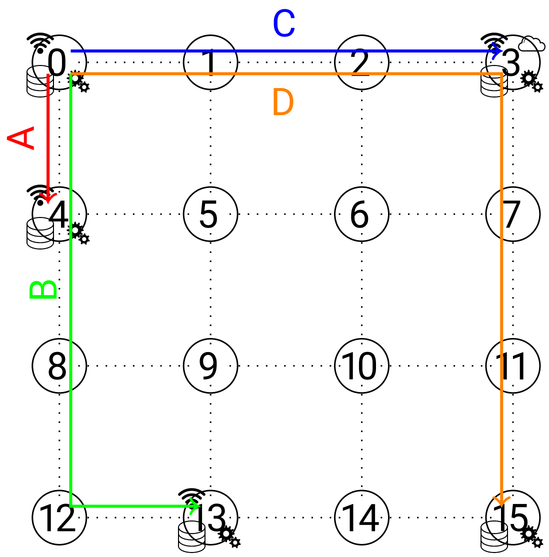
In Advance	On Demand
routing is done before forwarding	routing and forwarding is one mixed process

Embed Query Execution Plan in Topology

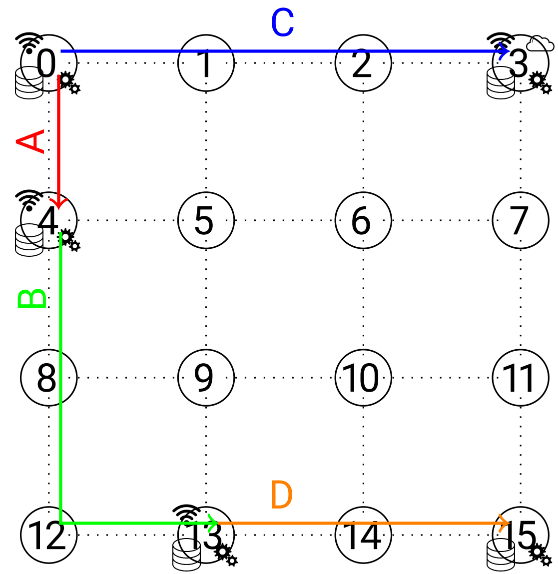


Inserting Data replicated due to indexing scheme

State-of-the-Art:



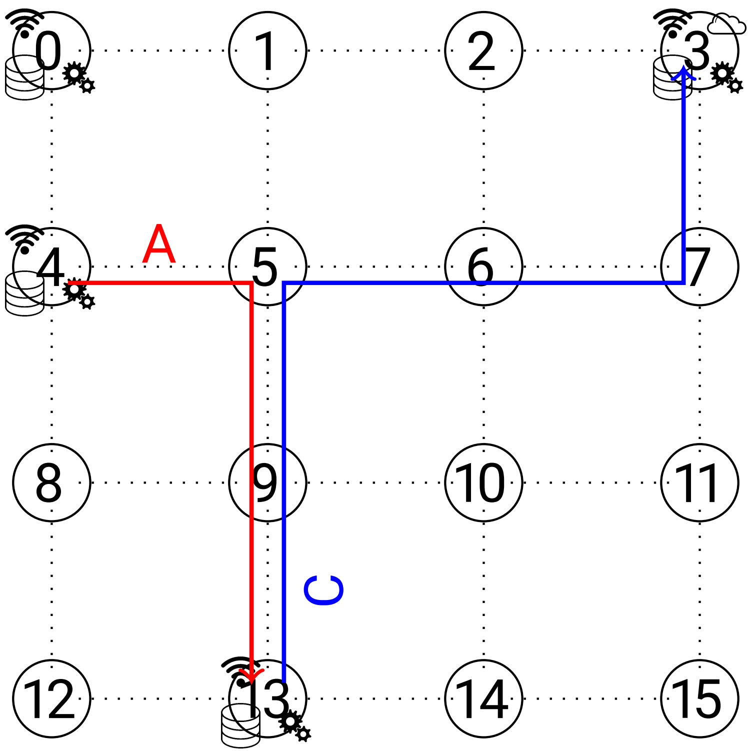
Multi-Cast:



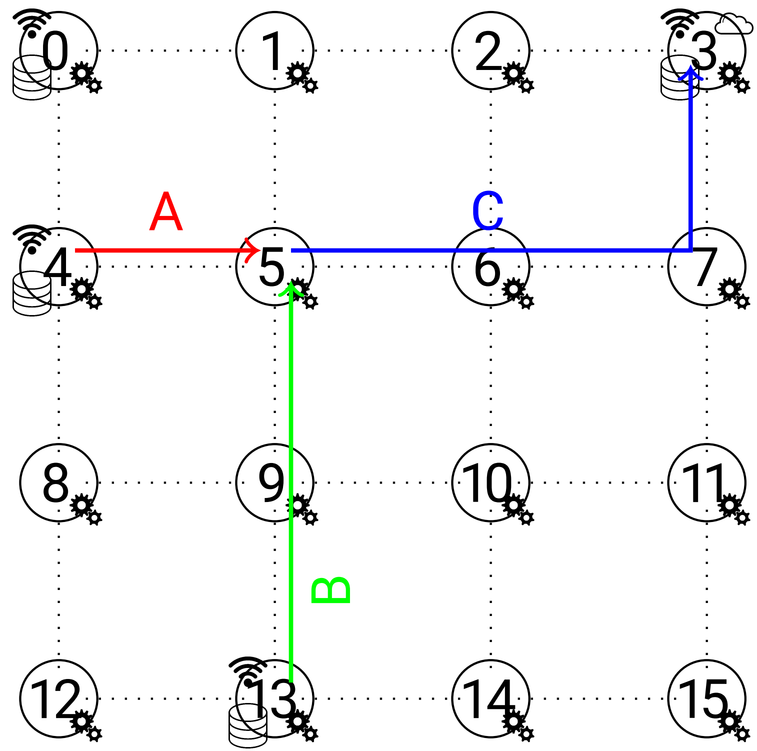
- network traffic reduction during insert by using
  - IPv6 Routing Protocol for Low-Power and Lossy Networks (**RPL**)
    $\longrightarrow$ All-Shortest-Paths (**ASP**): **17-48%**
  - **multi-cast**: **24%**
  - **additional devices**: **3%**

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

State-of-the-Art:



Combining Routing & Processing:

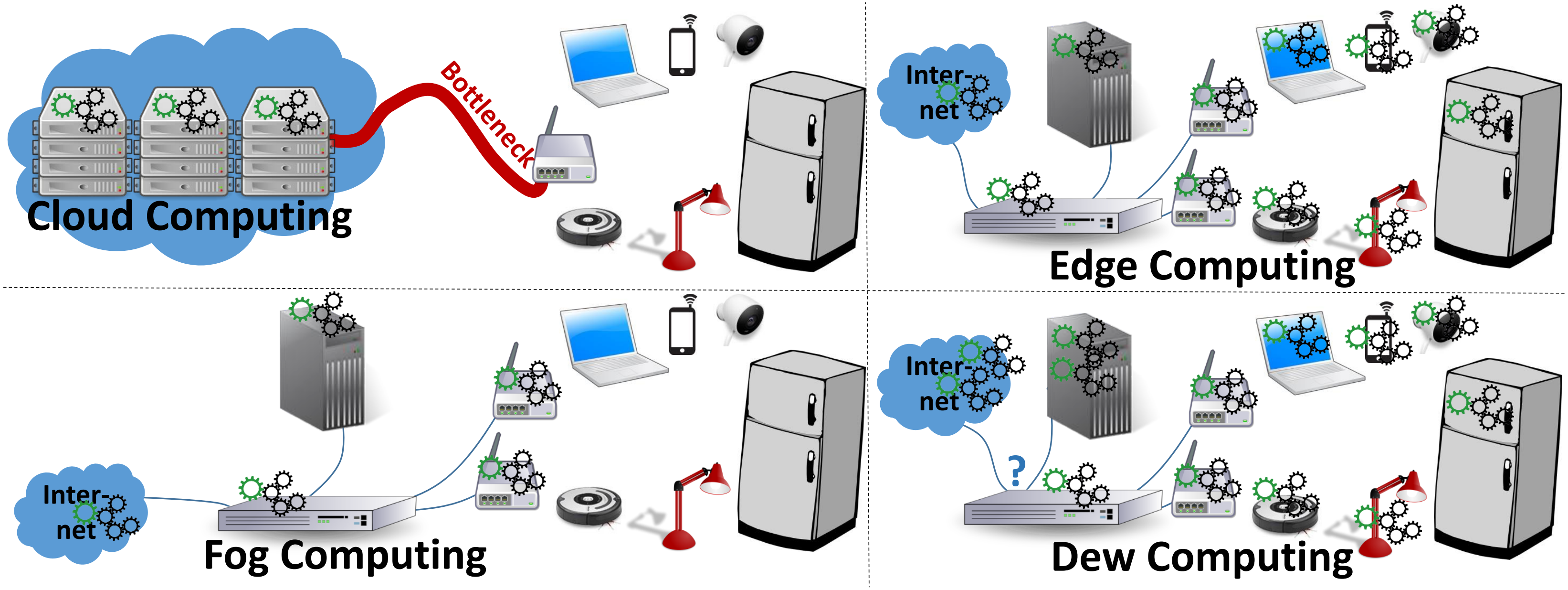


Thesis supervised by Prof. Groppe (S)IoT Database LUPOSDATE3000

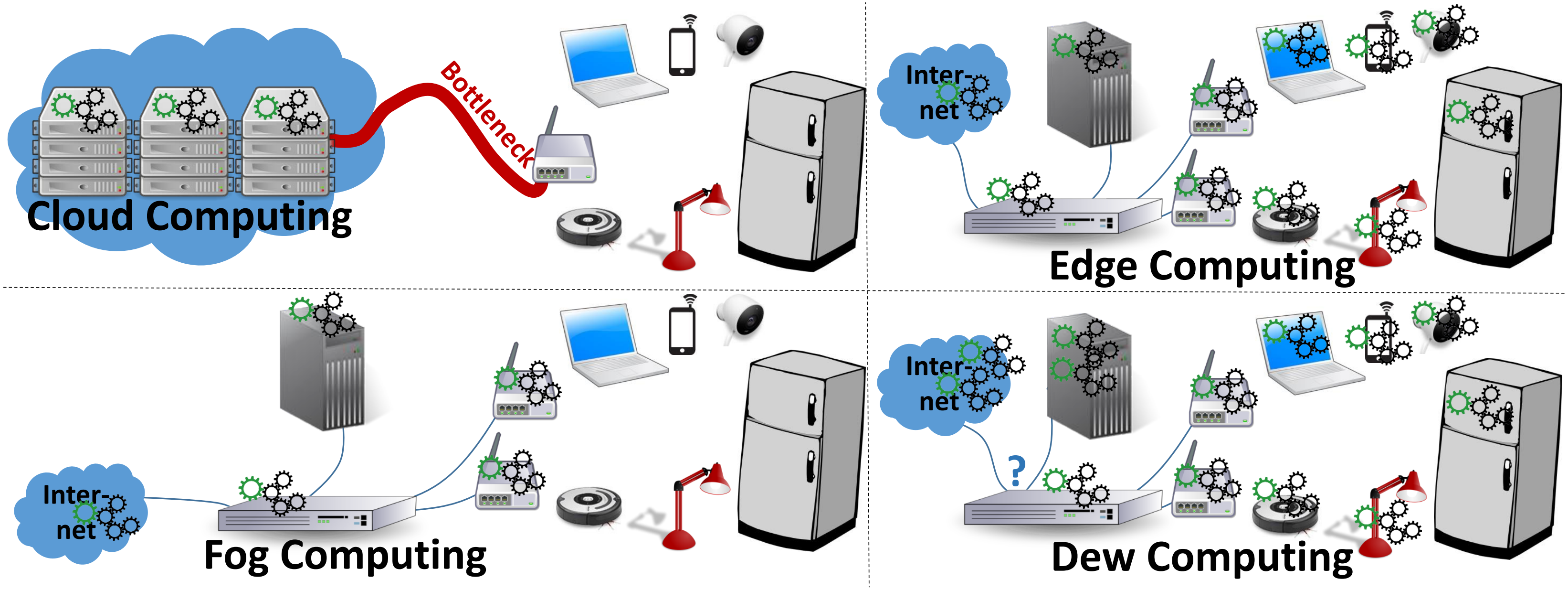
Wissenschaftl. Mitarbeiter: Benjamin Warnke/Simon Paasche

- Project BigSIoT (DFG, together with Stefan Fischer/ITM) (Warnke)
- External Promotion Bosch (Paasche)
- Combining (e.g., geo) routing protocols with query processing
- Compress data (in storage/messages) according to SHACL definitions, reformulate queries to directly work on compressed data
- Continuous Queries (e.g., redundant processing, efficient recovery after crashes, new types of windows for consistency checks)
- Distributed RDFS Inference
- Digital twins of machines in production
- Automatic Testing of Databases: randomized testing, generation of test queries, automatic minimization of test cases, ...
- Semantic Web Layer for NebulaStream (co-supervision with DIMA/Berlin)

IoT and Cloud as Platforms for Processing the IoT DATA

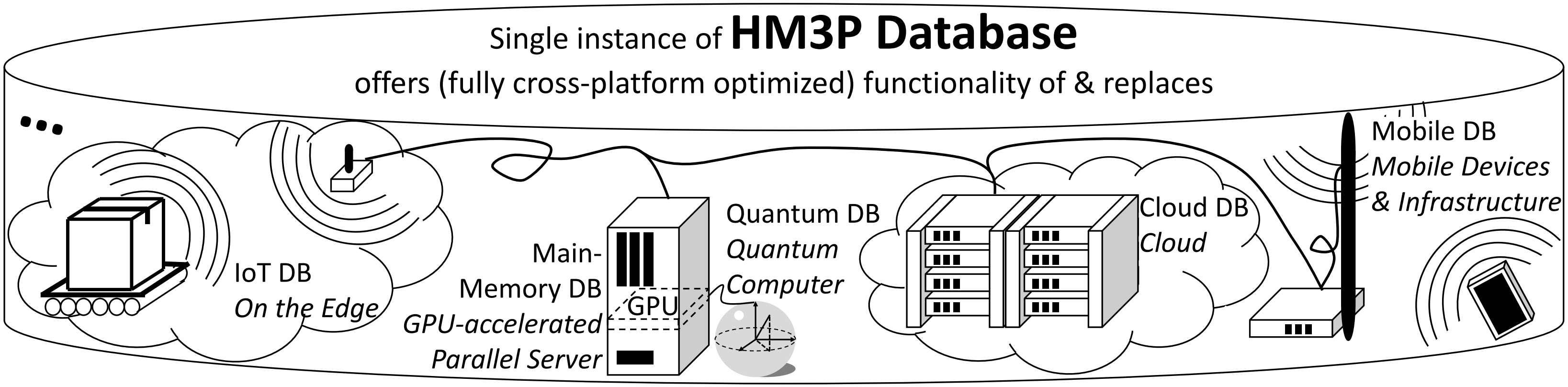


IoT and Cloud as Platforms for Processing the IoT DATA



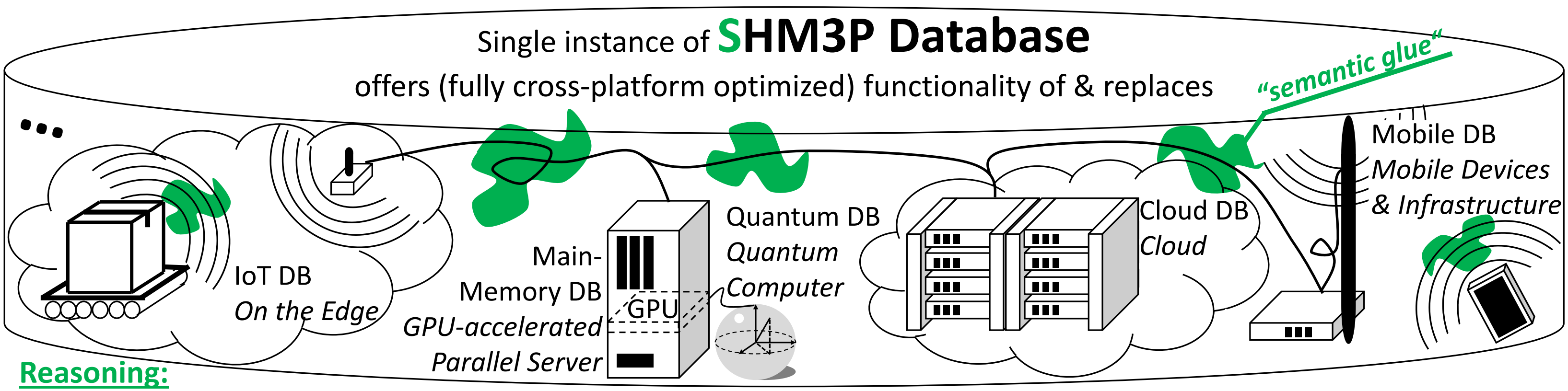
- ...but there are also **other platforms**!

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



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?

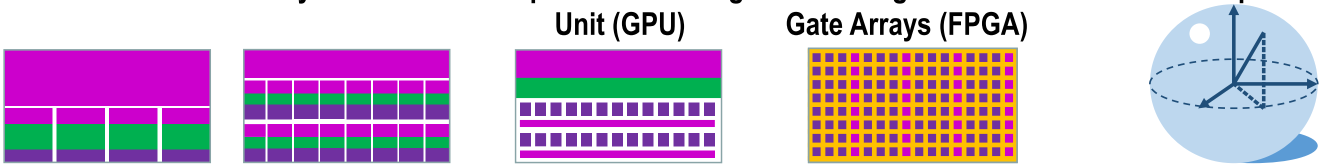
- Semantic Layer as glue between other models and platforms
- new challenges like integrating different types of reasoners in a transparent global reasoner

(+) Features of HM3P databases

(+) Easier data integration

(-) Performance issues may occur due to semantic layer

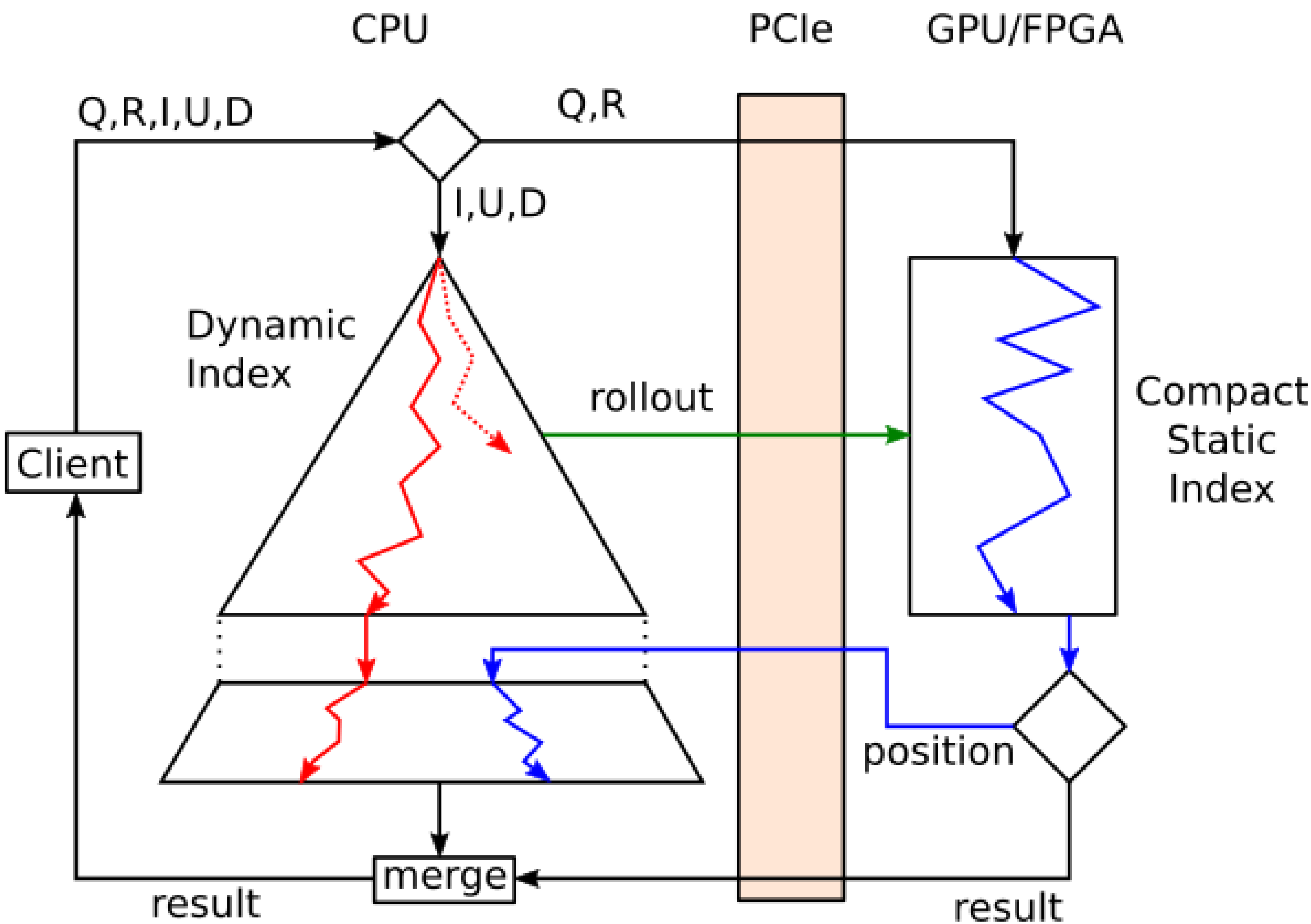
Architectures of Emergent Hardware



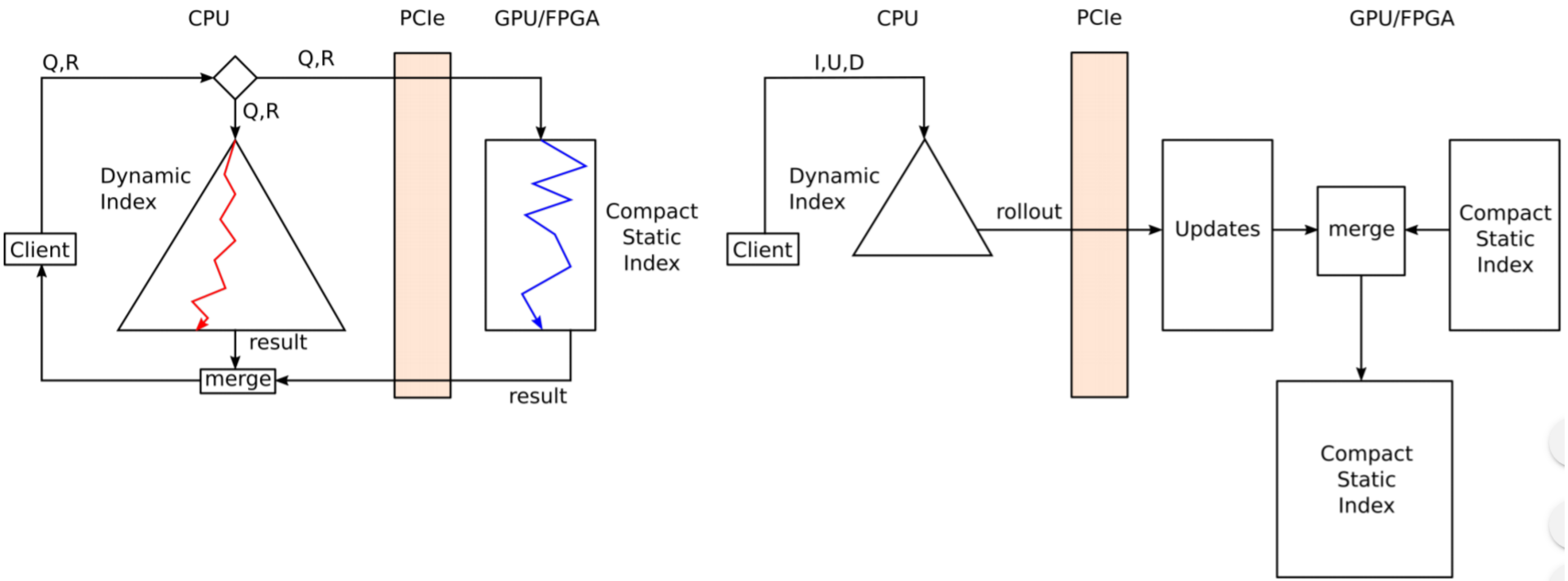
	Multi-Core CPU	Many-Core CPU	Graphics Processing Unit (GPU)	Field Programmable Gate Arrays (FPGA)	Quantum Computer/Annealer
Cores	~10	~100	~1000	~100 000	~50 / ~5000 qubits
Core Complexity	Complex (optimized for single thread performance)			Simple	Complex
Computational Model	MIMD + SIMD		SIMD	Data-Flow	Universal/Adiabatic Quantum computing
Parallelism	Thread and Data Parallel		Data Parallel	Arbitrary	Quantum Logic Gates/ Fluctuation (glob. Min.)
Memory Model	Shared	Distributed			Quantum Superposition
Power	150 W	200 W	250 W	50 W	25 KW
Database Op.	Query Optimization (Enumeration of Plans), Concurrency Control				
	Efficient Processing Of				
	Query Processing				

Legend: Computational Unit, Execution Controller, Interconnection Network, On-Chip Memory

Architectures for Hybrid Indices (Variant 1)



Architectures for Hybrid Indices (Variant 2)



Good News

- Framework for Hybrid Indices including Benchmark Suite available
  ⇝ Tobias Groth
- Offered topics deal with
  - integrating suitable/develop new indices in this hybrid approached

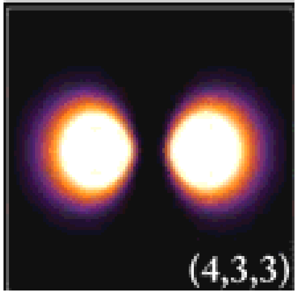
Thesis supervised by Prof. Groppe GPU Accelerated Index Structures

Wissenschaftl. Mitarbeiter: Tobias Groth

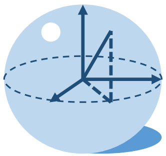
- Project Hybrid^2 (DFG, together with Prof. Pionteck (Magdeburg))
- Filter based on Hashing
  - Bloom: Paper Tutorial Code
  - Cuckoo: Paper Code 1 Code 2
  - Morton: Paper Code
  - Vacuum: Paper Code
- Search Trees
  - B/B+-tree: Paper Code
  - CSB/CSB+: Paper Code
  - B^{ed}-Tree: Paper Code
  - Masstree: Paper Code
- ...

Quantum Mechanics

- Very small particles and light behave differently from objects in normal life
- Mechanics of light and matter at the atomic and subatomic scale are described by quantum theory
  - forming the underlying principles of chemistry and most of physics
- Quantum theory has brought us the information age with its disruptive technologies of
  - transistors,
  - lasers,
  - nuclear power, and
  - superconductivity...
  - **...and now also quantum computers!**

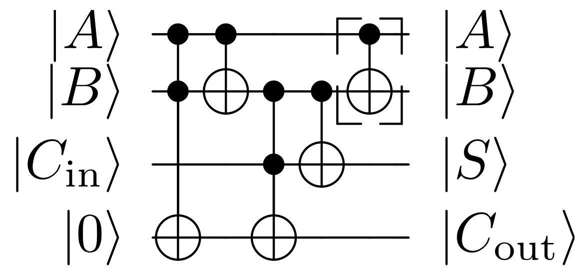


Wavefunctions of the electron in a hydrogen atom at different energy levels. Brighter areas represent a higher probability of finding the electron.

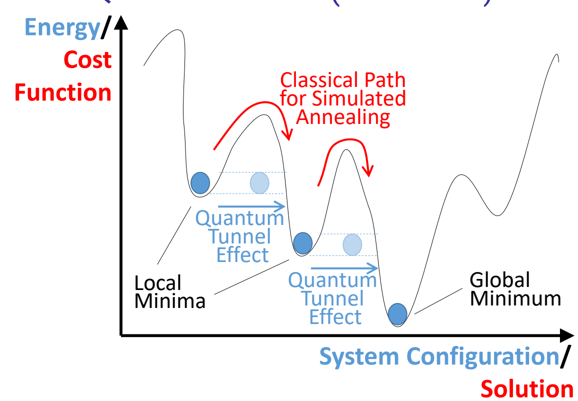


Quantum Computer

- use of quantum-mechanical phenomena such as superposition and entanglement to perform computation
- Different types of quantum computer, e.g.
  - Universal Quantum Computer
    - uses quantum logic gates arranged in a circuit to do computation
    - measurement (sometimes called observation) assigns the observed variable to a single value
  - Quantum Annealing
    - metaheuristic for finding the global minimum of a given objective function over a given set of candidate solutions
    - i.e., some way to solve a special type of mathematical optimization problem



Quantum Circuit (Full Adder)

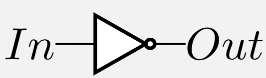


Simulated versus Quantum Annealing

Classical versus Quantum Computing

	Classical	Quantum
Information Unit	**Bi**nary Digi**t** (Bit): • basis of a 2-level system • can be in state 0 or 1	**Qu**antum **Bit** (Qubit): • basis of a 2-level quantum system • can be in state $\|0>$, $\|1>$ or in a linear combination of both states
Operation	Logic Gate: • performs on 1 or more bits to produce a single bit output	Quantum Logic Gate: • performs on 1 or more qubits to change the quantum state of a single qubit
Example Operation	NOT/Inverter: Digital Circuit: (see inverter circuit below)	NOT/Pauli$_x$-Gate: Quantum Circuit: (see circuit below) Alternatively: (see circuit below)

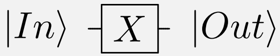
Digital Circuit:



In	Out
0	1
1	0

Quantum Circuit:

Alternatively:



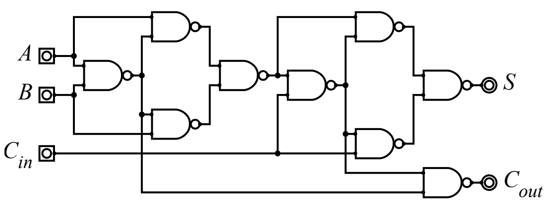
In	Out
$\|0>$	$\|1>$
$\|1>$	$\|0>$
$\frac{1}{\sqrt{2}}(\|0> + \|1>)$	$\frac{1}{\sqrt{2}}(\|0> + \|1>)$
$\frac{3 \cdot i}{5}\|0> + \frac{4}{5}\|1>$	$\frac{4}{5}\|0> + \frac{3 \cdot i}{5}\|1>$

Digital versus Quantum Circuits

	Digital Circuit	Quantum Circuit
Building Blocks	Logic Gates	Quantum Logic Gates

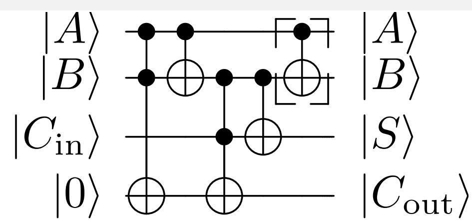
Full Adder Example

Digital Circuit:



consists of NAND gates

Quantum Circuit:



consists of Toffoli and CNOT gates[1]

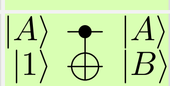
In- and Output Full Adder

Digital Circuit:

Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

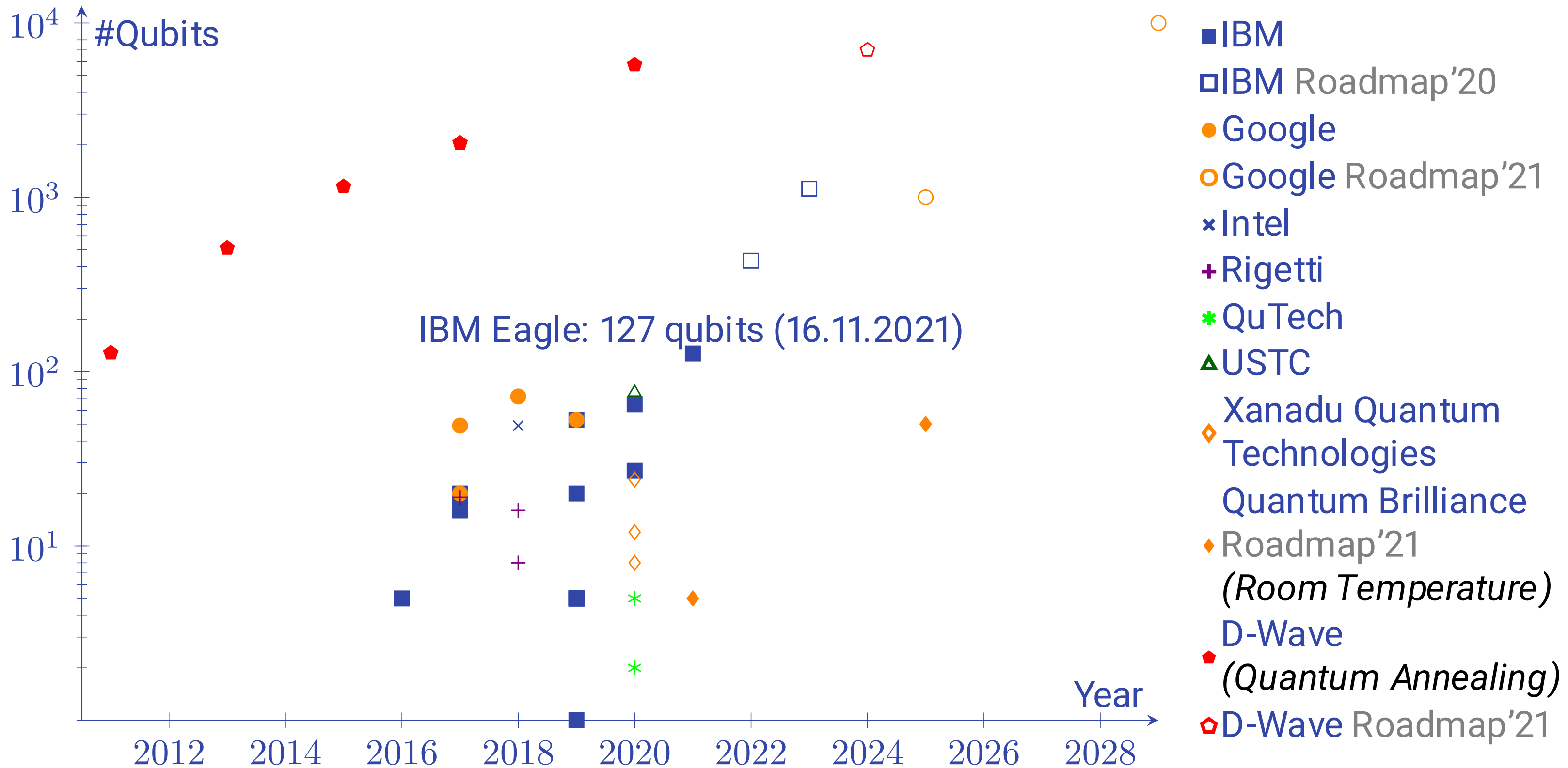
Quantum Circuit:

$|0>$ **and** $|1>$ **as input:** Output is $|0>$ and $|1>$ analogous to digital circuit.
Superpositions as input: Superpositions as output with corresponding probabilities for basic quantum states, e.g.:

A	B	C_{in}	C_{out}	S
$\frac{1}{\sqrt{2}}(\|0>+\|1>)$	0	0	**$ABC_{out}S$:** $\frac{1}{\sqrt{2}}(\|0000>+\|1001>)$	
$\frac{1}{\sqrt{2}}(\|0>+\|1>)$		0	0	1

[1] The dotted square marks a superfluous gate if uncomputation to restore the B output is not required. **[Feynman, 1986]**

Timeline of Quantum Computers



Potential of Quantum Algorithms

- **Quantum Algorithm Zoo** as example of collection of *important* quantum algorithms

Covered Years	1974-today
#Investigated References	430 (visited on October 2021)
#Algorithms	**64**
Speedups	**Superpolynomial: 31**, Polynomial: 27, Constant factor: 1, Varies: 3, Various: 1, Unknown: 1

Terminology:
α: positive constant
$C(n)$: runtime of the best known classical algorithm
$Q(n)$: runtime of the quantum algorithm
Superpolynomial Speedup: $C = 2^{\Omega(Q^\alpha)}$
Polynomial Speedup: otherwise

Very Important Quantum Algorithms 1/2: Shor's Algorithm[1]

- factoring integers in polynomial time
  - Depth of quantum circuit[2] to factor integer N: $O((log\ N)^2(log\ log\ N)(log\ log\ log\ N))$
  - superpolynomial speedup, i.e., almost exponentially faster than the most efficient known classical factoring algorithm (general number field sieve): $O(e^{1.9(log\ N)^{\frac{1}{3}}(log\ log\ N)^{\frac{2}{3}}})$
- Important for cryptography $\longrightarrow$ Post-Quantum Cryptography
- Most quantum algorithms with superpolynomial speedup like Shor's algorithm are based on quantum Fourier transforms (quantum analogue of inverse discrete Fourier transform)

[1] [Shor, 1994] [2] [Beckman et al., 1996]

Very Important Quantum Algorithms 2/2: Grover's Search Algorithm

- Black box function (oracle) $f : \{0, ..., 2^b - 1\} \mapsto \{true, false\}$
- Grover's search algorithm finds one $x \in \{0, ..., 2^b - 1\}$, such that $f(x) = true$
  - if there is only one solution: $\frac{\pi}{4} \cdot \sqrt{2^b}$ basic steps each of which calls f
    Let $f'(b)$ be runtime complexity of f for testing x to be true:
    $\Rightarrow O(\sqrt{2^b} \cdot f'(b))$
  - if there are k possible solutions: $O(\sqrt{\frac{2^b}{k}} \cdot f'(b))$
- Basis of many other quantum algorithms and applications

Lov K. Grover. A fast quantum mechanical algorithm for database search. *STOC'96*, 1996

Algorithms (used e.g. in Query Optimization) and their Quantum Counterparts

Query Optimization Approach	Basic Algorithm	Quantum Computing Counterpart
[S+79]	Dynamic Programming [E04]	[R19] [A+19]
[IW87], QA: [TK16]	Simulated Annealing [KGV83]	[J+11]
[MP18] [Y+20] [W+19] [O+19]	Reinforcement Learning [BSB81]	[S+21] [DCC05]
[GPK94]	Random Walk [BN70]	[ADZ93] [A+01]
[BFI91]	Genetic Algorithm [H92]	[W+13]
[TC19]	Ant Colony Optimization [CDM91] [DBS06]	[WNF07] [G+20]

This list is not complete...

Thesis supervised by Prof. Groppe
Open Challenges for QC for Databases

- Project QC4DB (BMBF, together with Quantum Brilliance)
- Are QC counterparts of basic algorithms used in query optimizations suitable for speeding up databases?
- What should be the properties of a quantum computer (e.g. #qubits, latencies of gates) to achieve certain speedups?
- How to combine classical and quantum computing algorithms to achieve good speedups with few qubits?
  (...for running database optimizations on current available quantum computers...)
- What other database domains besides query optimization benefit from quantum computers?
  (In short: those based on mathematical optimization problems, but also other...?)

Summary and Conclusions

- Different data models and their special features
  - ➡ Multi-Model Databases
- Different platforms and a need for different types of databases
  - Different features
  - ➡ Multi-Platform Databases
- Databases spanning over different platforms in operation (supporting multiple data models)
  - ➡ Hybrid Multi-Model Multi-Platform (HM3P) Databases
- Emergent Technologies like IoT, GPU, Quantum Computing
- We are offering **many topics for bachelor/master thesis**...
  - Please contact me: groppe@ifis.uni-luebeck.de