LiRA. LightKone Reference Architecture

Ali Shoker et al.

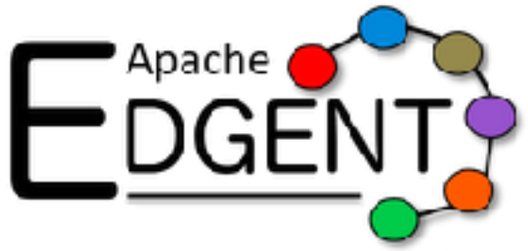
LightKone second official review
Brussels, Feb 2019

Roadmap

- ❑ Why a new Reference Architecture?
- ❑ LiRA Innovations
- ❑ Architecture View
- ❑ Component View
- ❑ Use-case Views

  - Multi-cloud metadata search (Scality)
  - Multi-master Geo-replicated Storage (Scality)
  - Distributed monitoring for community network (guifi.net)
  - Precision agriculture (Gluk)
  - NoStop RFID (Stritzinger)

OpenFog Consortium



Apache Foundation

Linux Foundation

FIWARE

ECC Consortium

Amazon Greengrass

Microsoft Azure IoT Edge

ETSI MEC

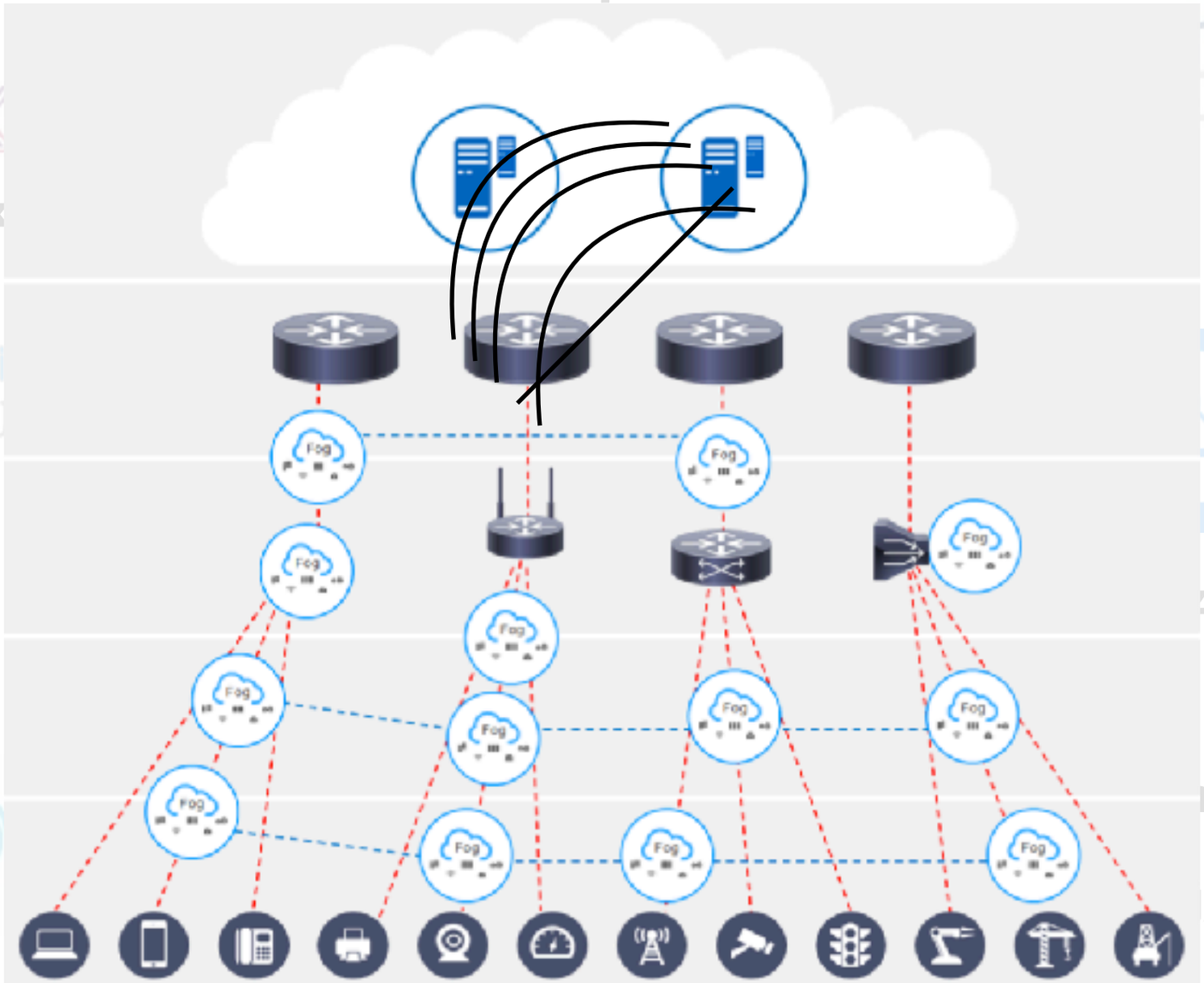
GeeLytics

NEC (Cheng, 2015)

SpanEdge

KTH (Sajjad, 2016)

- **Vague or undefined application-level data and communication aspects**
- **Data sharing is done across edge/fog network layers**
- **Example: OpenFog Visual security (surveillance) use case**

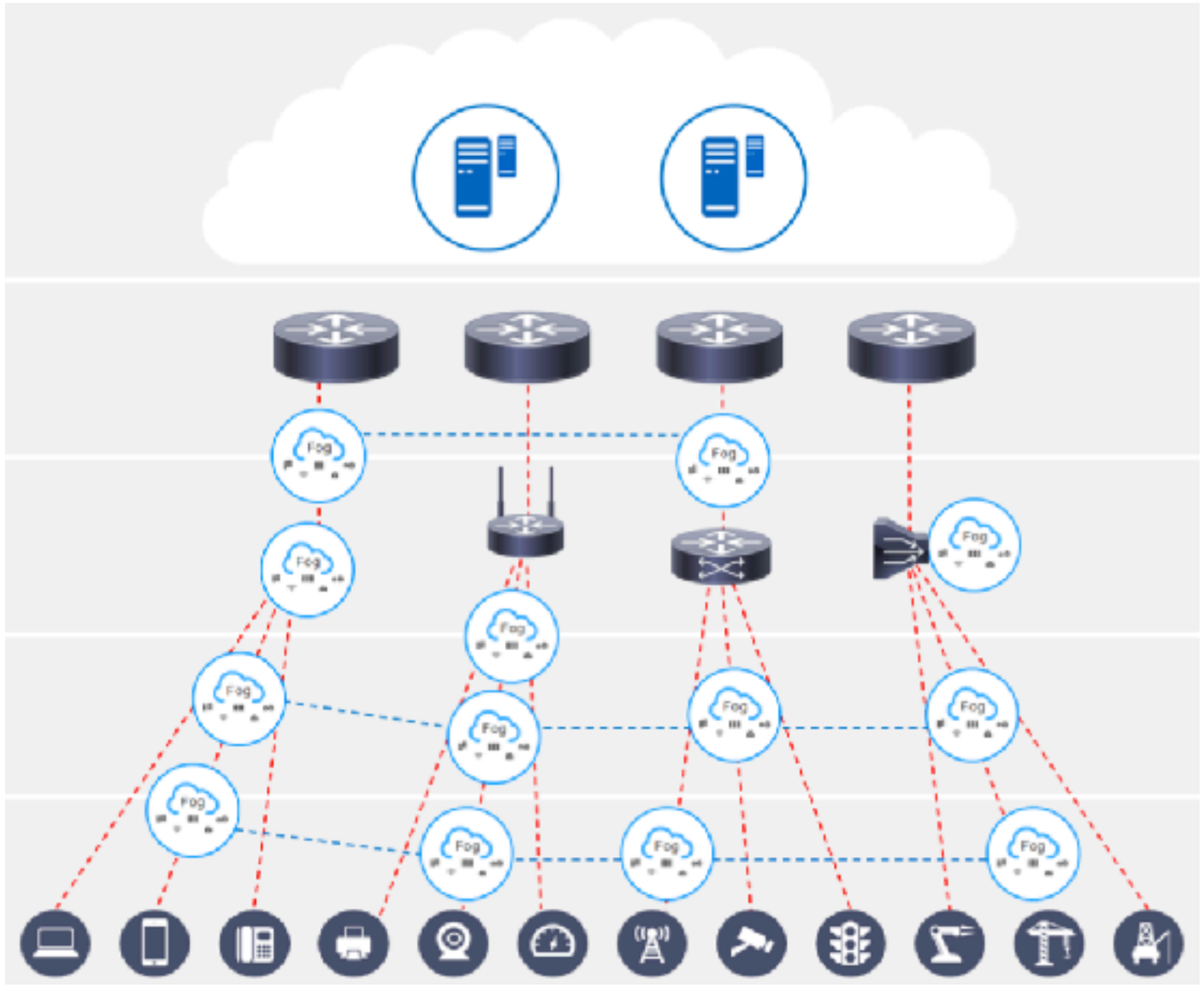


LiRA innovation

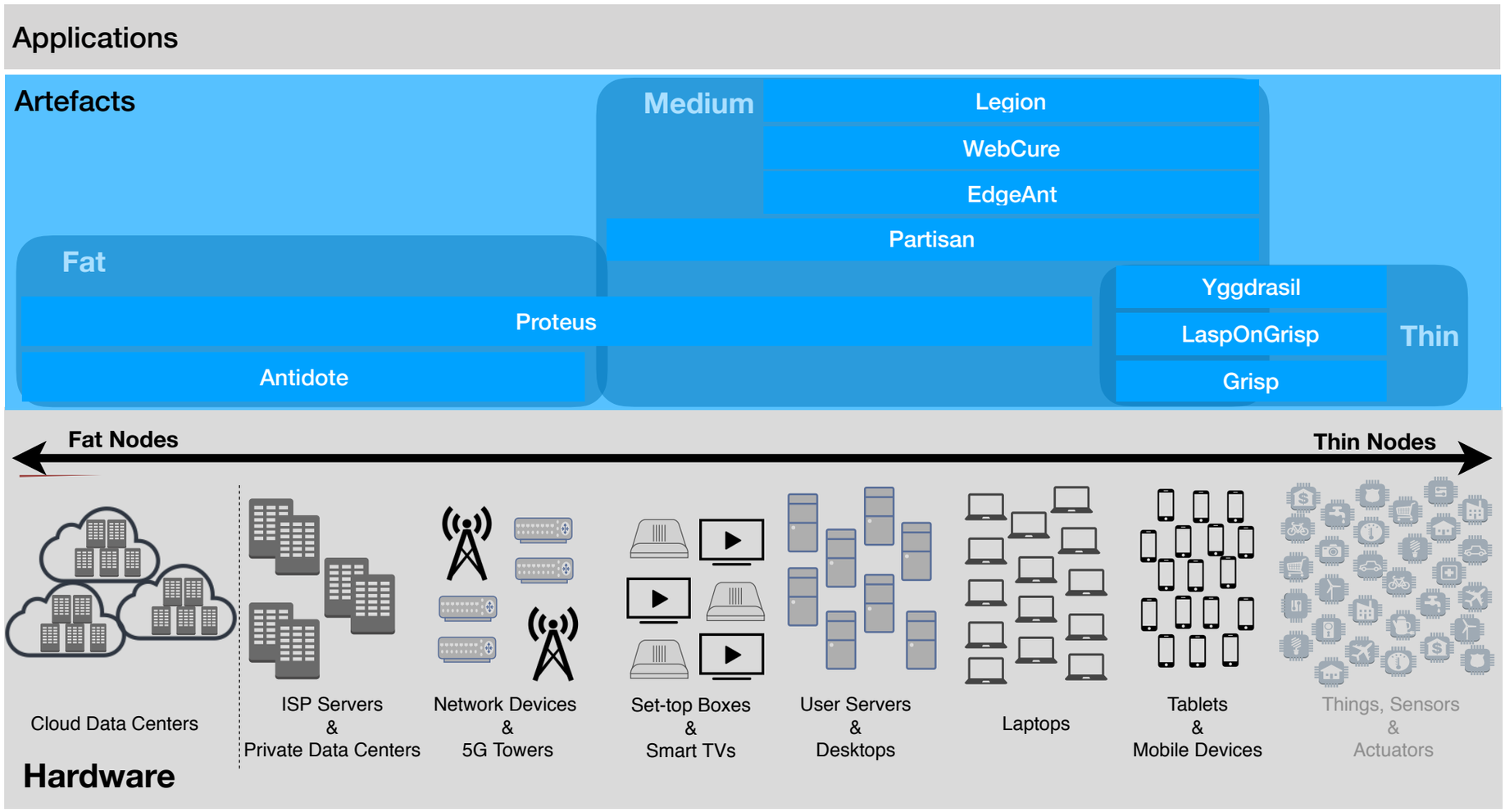
- **Convergent lateral data replication (sharing)**

- **Focus on application-level data and communication patterns**

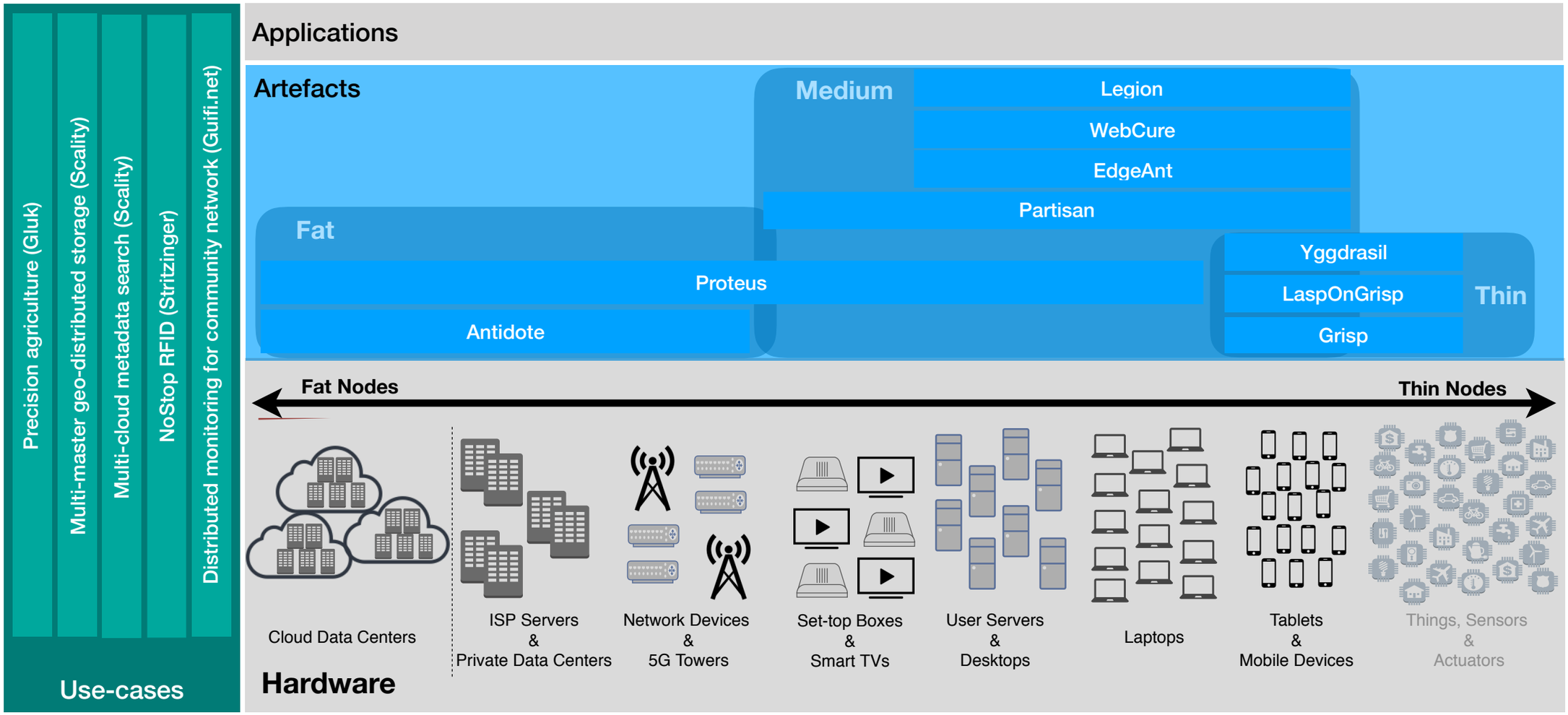
- **Classes: Heavy, Light, and Hybrid Edge**



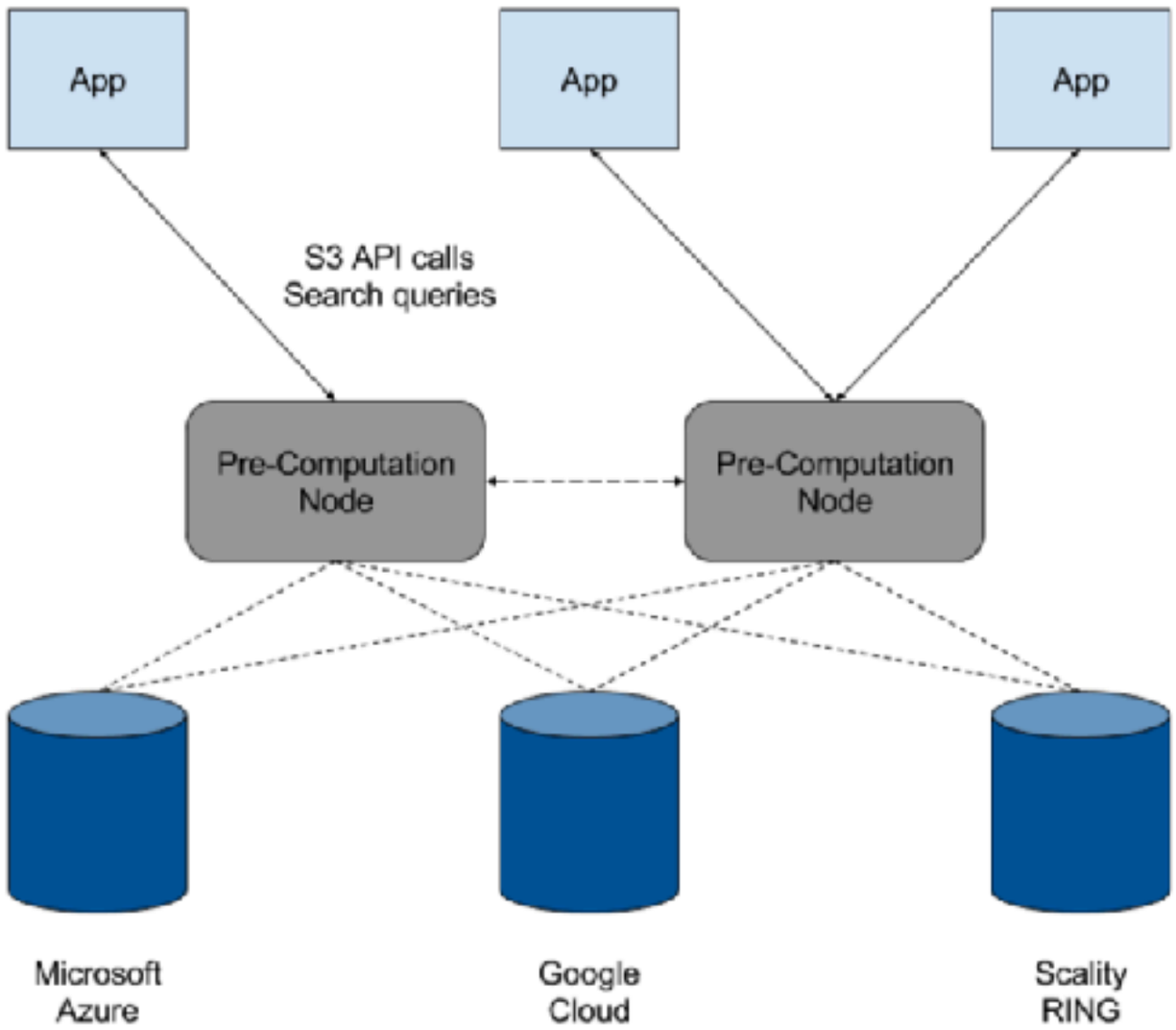
Architecture View



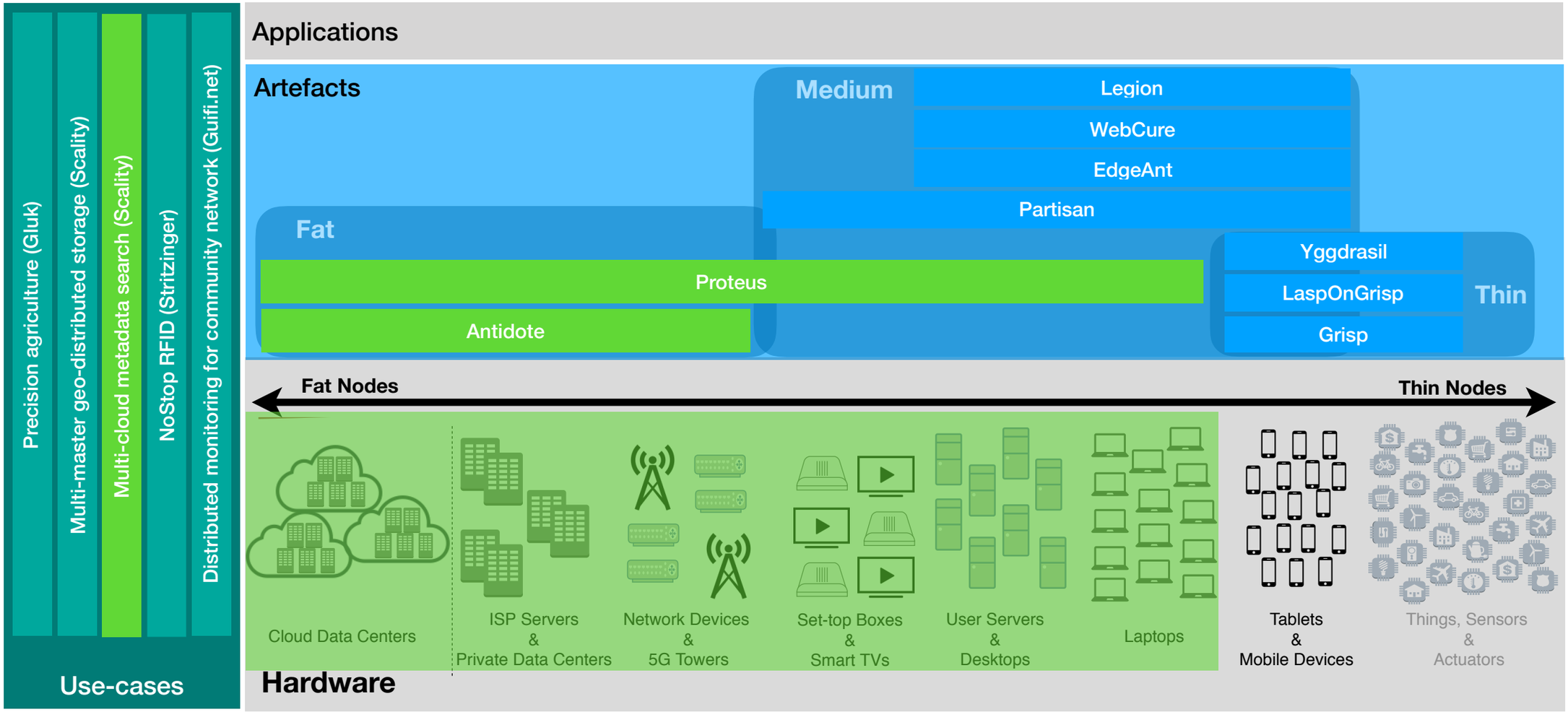
Use-Case View



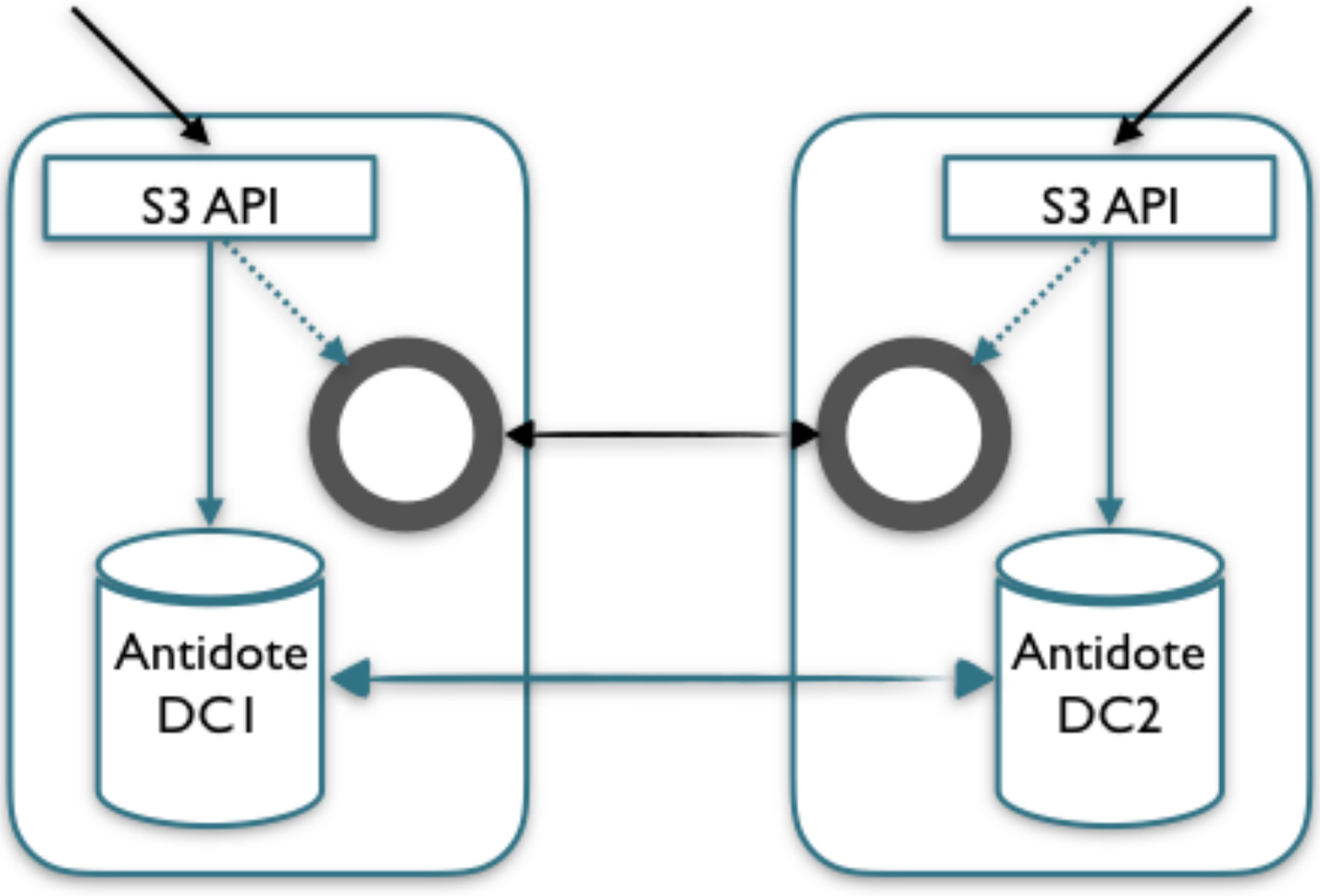
Use-Case View
Multi-cloud metadata search (Scality)



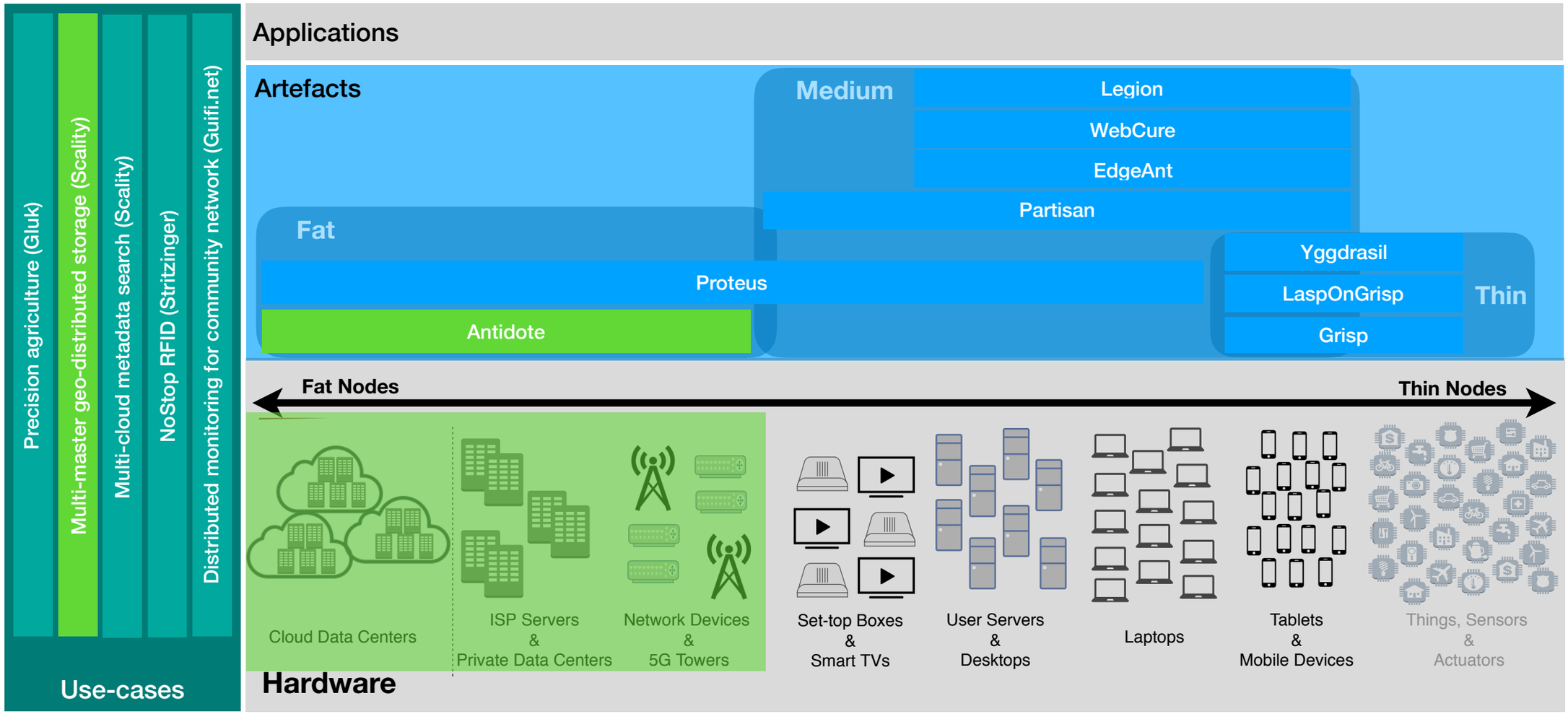
Use-Case View
Multi-cloud metadata search (Scality)



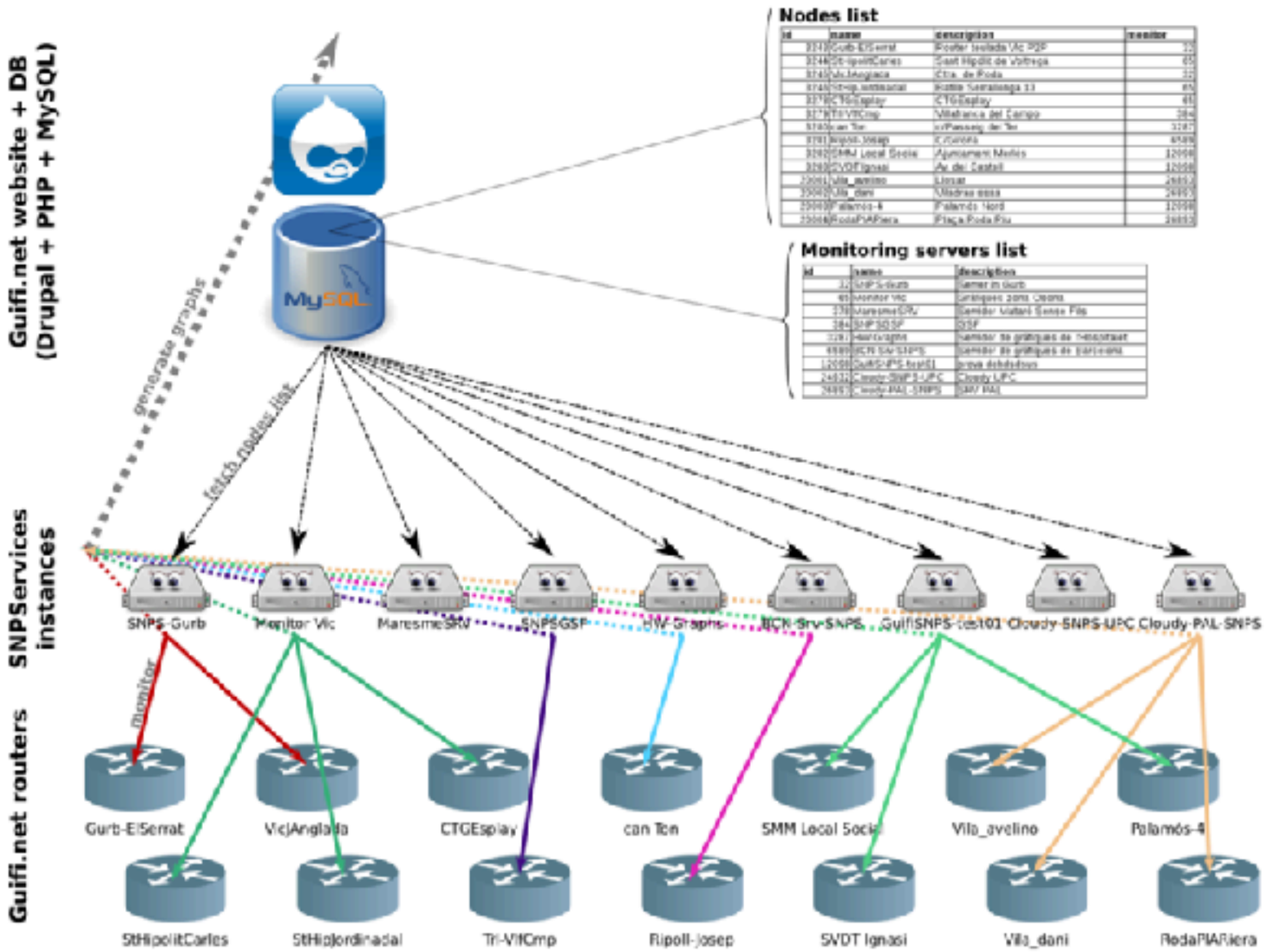
Use-Case View
Multi-master Geo-replicated Storage (Scality)



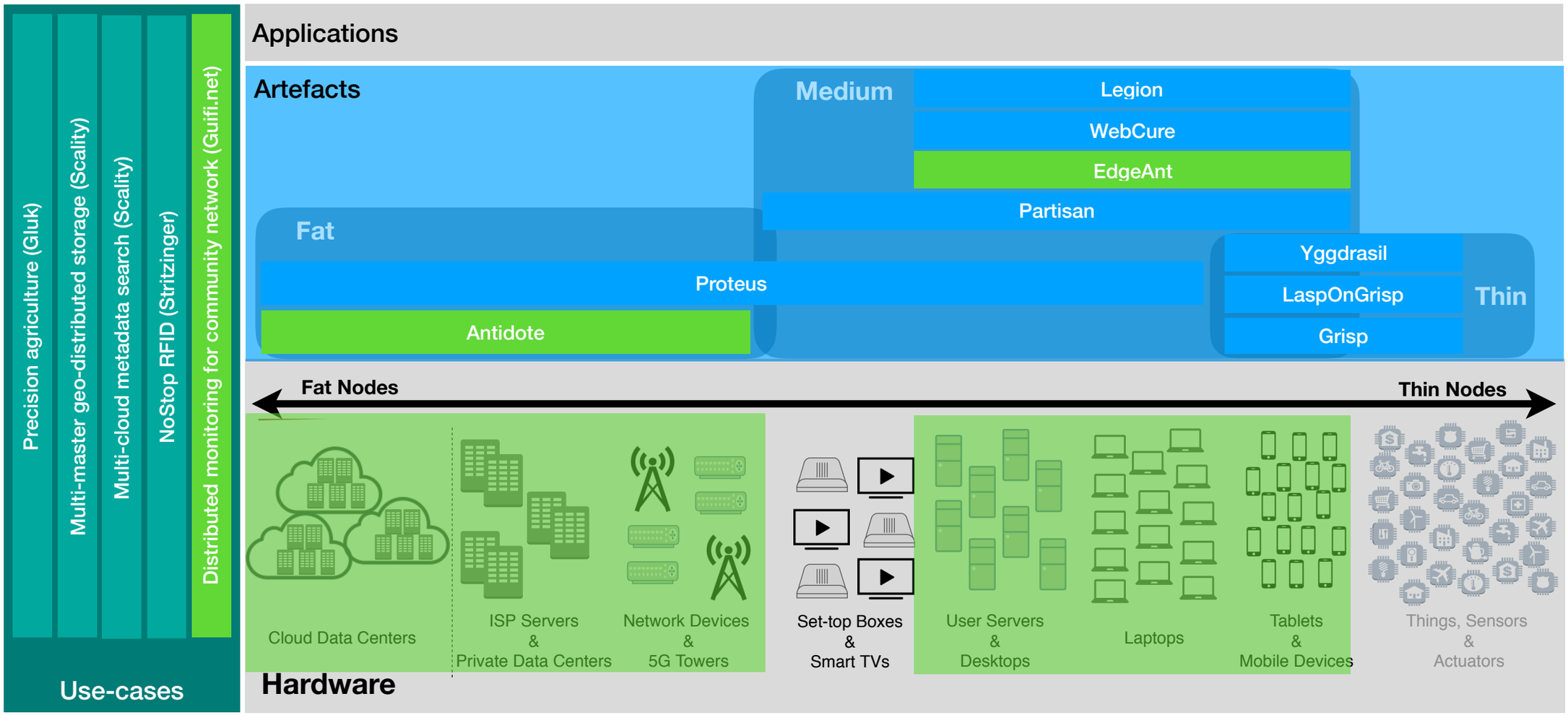
Use-Case View
Multi-master Geo-replicated Storage (Scality)



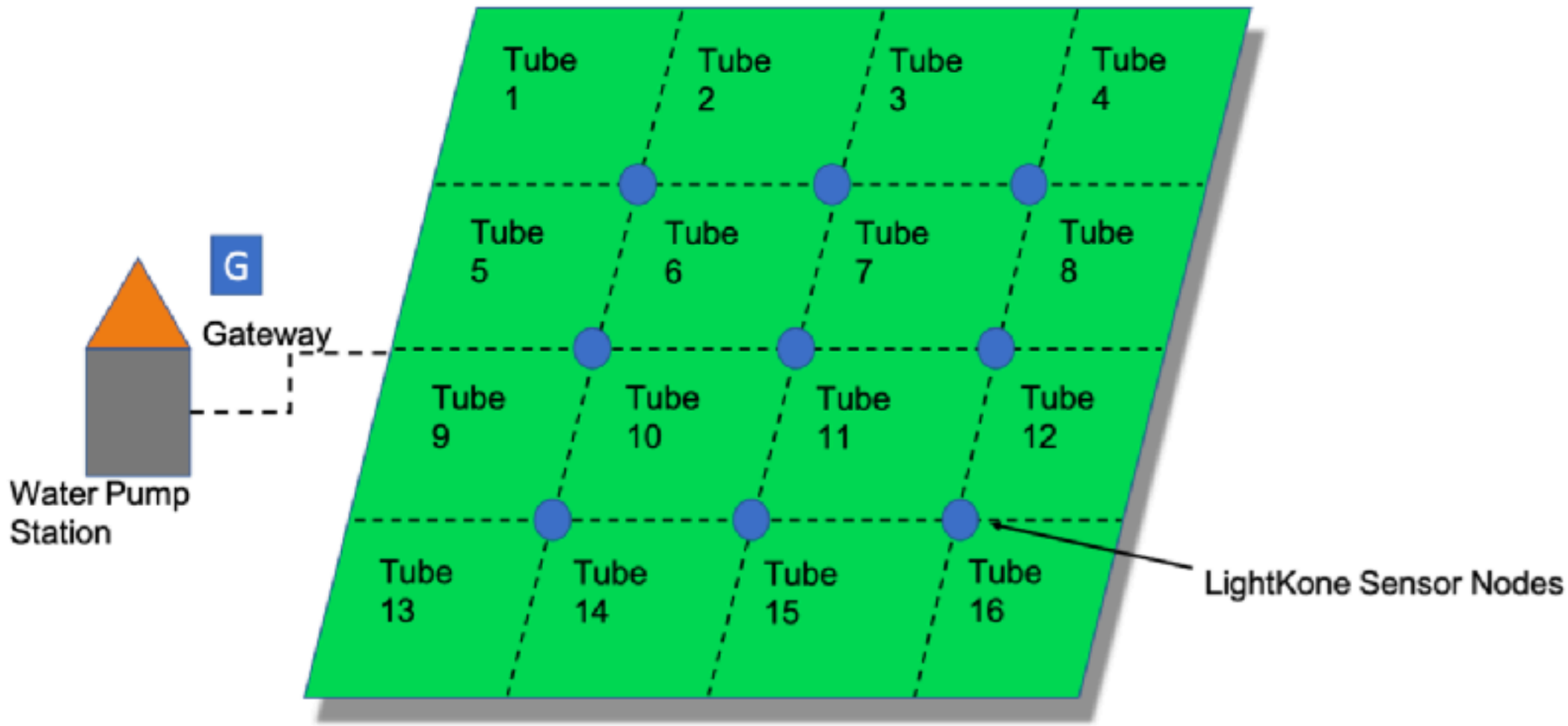
Use-Case View
Distributed monitoring for community network (Guifi.net)



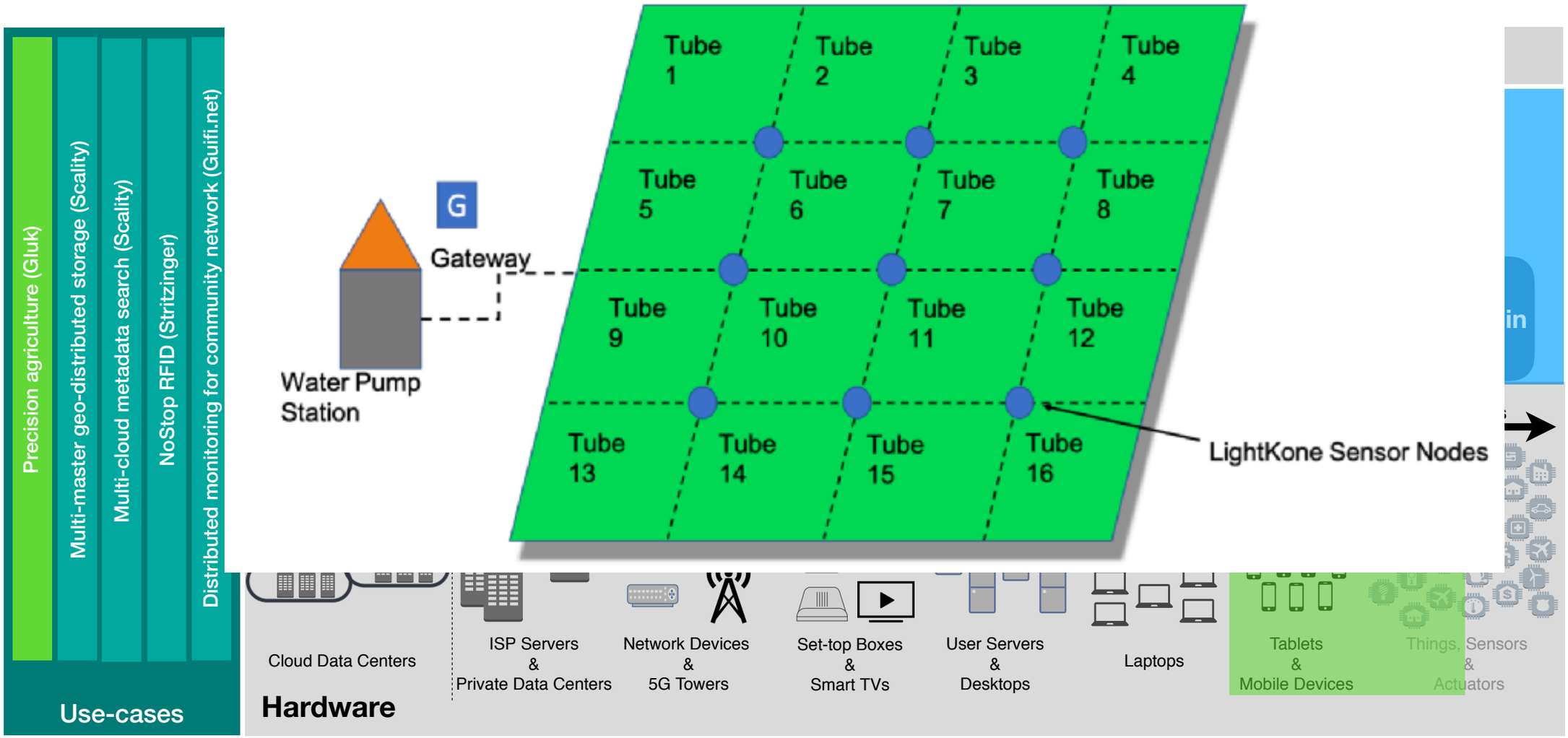
Use-Case View
Distributed monitoring for community network (Guifi.net)



Use-Case View
Precision agriculture (Gluk)



Use-Case View
Precision agriculture (Gluk)



NoStop RFID (Stritzinger)
—maybe discard this!

Future Work

- ☐ Extend LiRA with more artefacts if needed
- ☐ Adjust the spectrums
- ☐ Improve interfacing
- ☐ Dissemination

Backlog

Artefact	Description	Previous SOTA	Contribution	Read more
AntidoteDB	A highly available geo-distributed database	Geo-replicated databases with different consistency semantics, typically either weaker (EC) or stronger (Serializability within shards)	Causal transactions + CRDTs	D6.1
WebCure	Client-side data replication for web applications using AntidoteDB as backendTODO: Annette	Read-only caches / roll back on updates on conflict	Simplified programming model with conflict resolution on CRDTs	D6.2
Legion	A framework for extending web applications to the edge, by running code in the client devices that interact directly.	Systems that support disconnected operation, but no peer-to-peer synchronization; Mobile systems that support peer-to-peer interaction, but that are not designed to support web applications.	Simple programming model for extending web application with peer-to-peer synchronization.Big delta CRDTs.Model for interacting with cloud services.Security mechanisms.	D5.1
EdgeAnt	A consistent, mutable cache at the edge. Data is backed up in Antidote. EdgeAnt supports the same API as Antidote, and guarantees the same TCC+ consistency. A cache can transparently disconnect and reconnect to any data centre. Ongoing work: (i) A client has the option to place any individual computation, either at the edge or in a data centre; both guarantee the same consistent view of data. (ii) Co-located EdgeAnt clients can collaborate in a group, even disconnected from the infrastructure, and can migrate between groups.	Edge caching for immutable data; or non-AP "sticky" or ad-hoc caches with ill-defined guarantees	Consistent, mutable AP cache. Uniform (DC to edge) AP guarantees Client can migrate Place computation @edge or @centre P2P group communication Client can change groups	D6.2

Yggdrasil	Framework for designing distributed protocols for ad-hoc networking.	Frameworks for developing distributed protocol, but no specific one for wireless ad-hoc networking.Multiple protocols for ad-hoc networking.	Simple programming model for defining new protocol, hiding the complexity of configuring wireless radios and exchanging messages among multiple communication.parties.	D5.1
Proteus	A geo-distributed framework for analytics computations on federated data stores. Proteus maintains materialized views and performs stateful data-flow computation. Admins place computation and data according to SLA considerations.	Apache Spark,Distributed search for federated clouds, Federated query processing on linked data,Lasp??	Bidirectional data-flow computations using materialized views stored as CRDTs. Modular distributed architecture that enables flexible data and computation placement in geo-distributed systems.	D6.2
Grisp	A Unikernel approach running the Erlang VM directly on smaller hardware without intervening operating system level. There is a software stack that allows for mixed critical systems with hard and soft real-time parts. A evaluation and development board for this was developed outside the project and provided to partners.	Running Erlang on Embedded Linux like operating systems. Soft real-time only.	Erlang on smaller IoT devices which wouldn't be able to run Linux. Erlang as part of mixed critical systems. Preparation for allowing hard real-time Erlang processes.	D5: Chapter 3.4
LaspOnGrisp	Reliable key/value store running on network of Grisp boards, allowing applications to run directly on the sensor boards. Reliable data storage based on CRDTs; reliable communication based on hybrid gossip (Partisan).	SOTA edge applications do not run on sensor networks, but on gateways that manage these networks.	Resilient data storage and resilient communication directly on sensor networks.	D4.2