Global Data Plane
The Cloud is not enough: Saving IoT from the Cloud & Toward a Global Data Infrastructure

PRESENTED BY – MEGHNA BAIJAL
Why is the Cloud Not Enough?

Currently, peripherals communicate directly with the cloud over the network and with each other through web services.

But there are problems —

1. Privacy and security
2. Scalability
3. Latency
Why is the Cloud Not Enough?

4. Bandwidth
5. Quality of Service
6. Durability
7. Currently application level gateways required which leads to stovepipe solutions
IoT Applications

Two Types of IoT Applications -

1. Ambient Data Collection
   - Sensors installed, Example — NeighborhoodWatch

2. Real Time Applications
   - Low Latency requirements, Example — a fan being sensor controlled, PreHeat

What’s Common?

It is almost always Time-Series Data
“The Global Data Plane (GDP) provides a data-centric glue for swarm applications”
The Global Data Plane

A Narrow Waist

Home Control, Smart Office, Follow-me Display, ...

File System, SQL, Key-value, ...

TCP/IP, UDP/IP, ...

Ethernet, Wi-Fi, Bluetooth, 802.15.4, AVB

Application

CAAPI

Log

Connection

Physical
The Log Interface

- Lightweight Data Structures — ordered list of records
- Durable
- Time Series
- Immutable
- Single writer
- Append only
- Multiple readers (Random Access or Subscription)
- Chunked and Distributed
- No fixed location but are migrated
- Each sensor has its own log
- Each Actuator reads from a unique log
Security & Key Management

1. Every Log is encrypted - Combination of Key Signatures and Merkle Trees
2. Writer gets its public signature key at the time of log-creation
3. A writer signs each append.
4. Log server validates it
5. The authorized readers have the decryption key

*The paper has a good example of how such an encryption might work.*

*However, the authors claim that this is not enough.*
Location Independent Routing

1. Logs are stored on Log-servers
2. Packets are routed using DHT
3. The location is advertised to the routing layer
4. And each log is chunked.
Multicast Trees

Useful when the input data grows a lot
- When there are multiple subscribers to the same data
- Use existing routing techniques to reduce effective bandwidth
- Store some useful routes
The Control Plane

1. Services that provide policy enforcement
   ◦ Example, controls replication
   ◦ Example, checks read write authorities

2. Separation of Mechanism from Policy
Flat Address Space

1. Logs, Clients, Servers all named using a 256 bit identifier.

2. Can be derived from a cryptographic hash of the owner’s public key and metadata

3. This is important so a log can be located even if replicated or migrated
**CAAPI (Common Access API)**

1. Logs are sufficient to implement any convenient data storage repo
   - Key value store
   - Filesystem
   - Database operations

2. CAAPI exposes minimal methods —
   - Read
   - Append
   - Subscribe
(a) Clients writing to Logs
(b) Chunking logs to store them
(c) Multiple Subscribers
(d) Log Migration
The Design
GDP Overcomes some Pitfalls of the Cloud

1. **Flexibility** —
   - Log is minimum but complete to build apps
   - Multiple logs can be combined as required

2. **Access Control, Privacy and Security** —
   - Encryption
   - Only Append is permitted with a writer’s key
   - Data Integrity — no traditional consistency problems (single writer!)

3. **Durability and Performance** —
   - Secure and consistent Replication
   - Chunking of Data
   - Flat Address Space
GDP Overcomes some Pitfalls of the Cloud

4. **Scalability** – DHT
5. **Latency** – Log Migration
6. **Bandwidth** – Multicast Tree
7. **Some Fault Tolerance** – Replication (behave like Content delivery systems)
**Strengths**

1. **Device virtualization**: A seamless interplay of heterogeneous components
2. **CAAPI**: a Common Interface for easier App Development
3. Reduces the load of security enforcement on device and app development
4. **Querying** Data becomes easy as it is time series data.
5. Latency benefits from **Pre-Fetching**
6. Immutable data makes **Caching** easy
Weaknesses

1. Absolutely no numeric results!
3. Currently fails the test of wide-scale deployment
4. Log formats must be well-specified and complied with.
5. Key management needs to be bettered
6. Need to work on routing and delays
7. Assumption that the devices have cryptographic keys for signing and encryption or that they use a gateway – is this a valid assumption?
Main Takeaways

1. GDP promises to make the Cloud more IoT friendly.
2. Such a Data Centric Model can be used beyond the world of IoT.
3. Most IoT applications can be modeled as time-series logs (Good discussion in ‘BOLT’).
4. Such a framework is needed to better exploit Fog Computing ideas.
5. The infrastructure discussed has a lot of open ends which leaves room to creatively add more flexibility and benefits.
The Future

1. Distributed Swarm Infrastructure

2. Global Data Plane — Implementation In progress
   - Release version 0.7
   - Plan to Release version 1.0 this year
   - Can track some features and bugs on the UCB website
References

2. https://swarmlab.eecs.berkeley.edu/research/distributed-swarm-infrastructure
6. https://www.usenix.org/node/179754
Questions?
Thank You!