

Summary of Research and Proposals

DSP Lab 2017

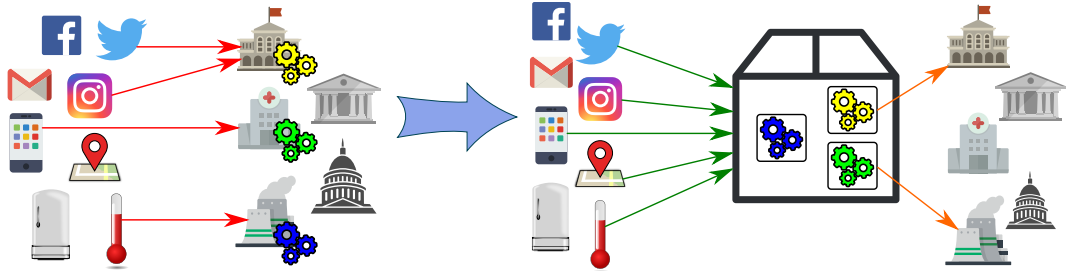
Yousef Amar



2017-10-02

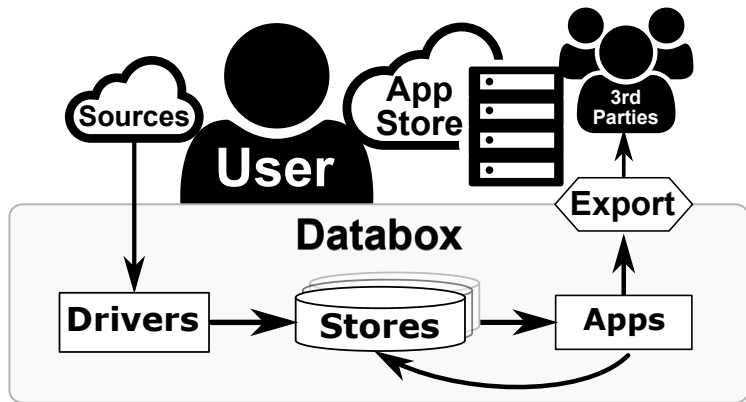
Research Context

The Databox Platform



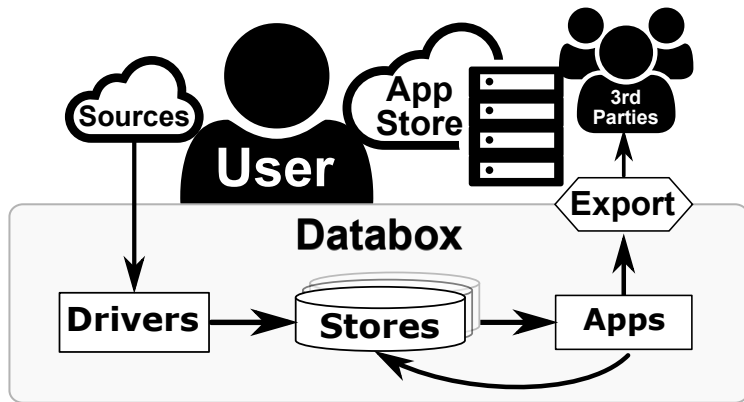
Research Context

The Databox Platform



Research Context

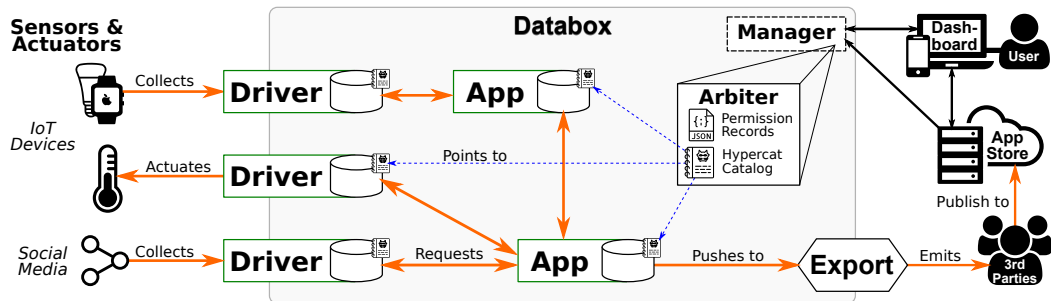
The Databox Platform



How can we design safe, scalable access control systems with arbitrary restrictions in this context?

Research Context

The Databox Platform



Implementation

The Route

- ▶ Triad of *target*, *path*, and *method*
- ▶ The container as a host
- ▶ RESTful APIs for all operations
- ▶ Direct mapping of HTTP methods to CRUD functions
- ▶ Per-route granular permissions

```
{  
  "target": "smartphone-store",  
  "path": "/accelerometer/ts/latest",  
  "method": "POST"  
}  
  
{  
  "target": "smartphone-store",  
  "path": "/(sub|unsub)/gps/*",  
  "method": "GET"  
}
```

Implementation

Delegated Authorization

- ▶ Google Research: Macaroons
 - ▶ A standard similar to signed cookies
 - ▶ Can be attenuated by “caveats”
 - ▶ Embedded permissions
 - ▶ Minting and verification can be separated through shared secret keys

```
target = smartphone-store  
path = /(sub|unsub)/gps/*  
method = GET  
time < 1489405851417
```

```
target = smartphone-store  
path = /light/ts/range  
method = GET  
startTimestamp >= 1489405234352  
endTimestamp <= 1489405259525
```



Implementation

Resource Discovery

- ▶ API for describing APIs
- ▶ Directory servers
- ▶ Many competing standards
 - ▶ Resource Description Framework (RDF)
 - ▶ Web Application Description Language (WADL)
 - ▶ Web Services Description Language (WSDL)
 - ▶ eXtensible Resource Descriptor (XRD)
- ▶ Subject-predicate-object style prevalent
- ▶ Different formats and applications — XML for REST, SOAP, OpenID

Implementation

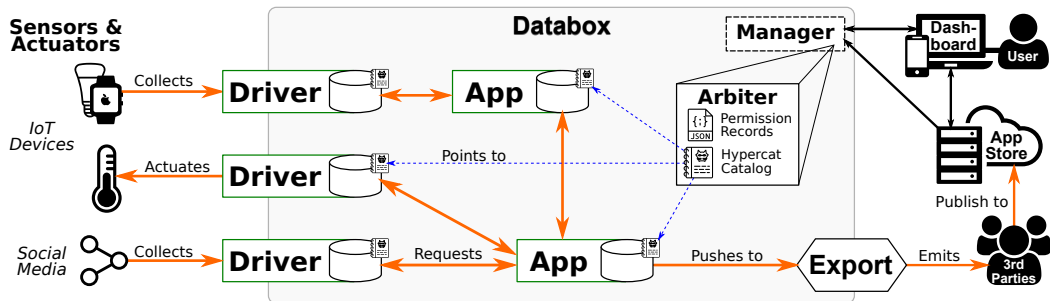
Resource Discovery

- ▶ Hypercat: Recently joined BSI Group
- ▶ IoT-first specification design
- ▶ JSON/REST over XML/SOAP
- ▶ Only cataloguing; ontologies and authorisation extensible
- ▶ Discoverability vs accessibility
- ▶ Catalogues can be nested, allowing decentralisation and distribution

```
{
  "catalogue-metadata": [{
    "rel": "urn:X-hypercat:rels:isContentType",
    "val": "application/vnd.hypercat.catalogue+json"
  }, {
    "rel": "urn:X-hypercat:rels:hasDescription:en",
    "val": "A Databox Store"
  }],
  "items": [{
    "href": "http://some-store/light",
    "item-metadata": [{
      "rel": "urn:X-hypercat:rels:hasDescription:en",
      "val": "Light Datasource"
    }, {
      "rel": "urn:X-databox:rels:hasVendor",
      "val": "Databox Inc."
    }, {
      "rel": "urn:X-databox:rels:isActuator",
      "val": false
    }
  ]
}]
}
```

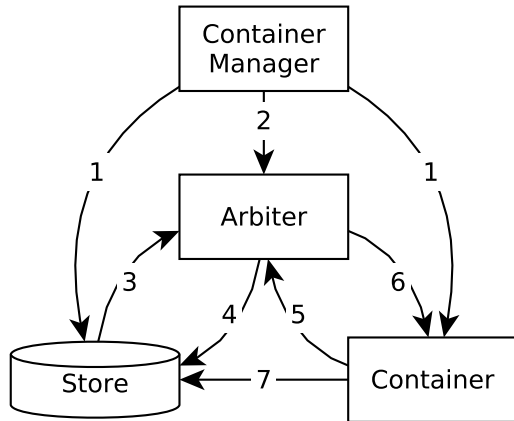
Implementation

The Arbiter



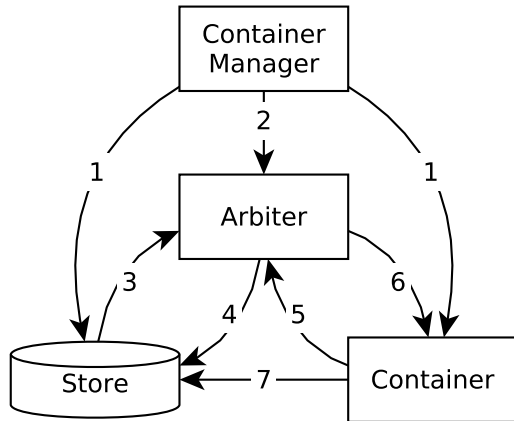
Implementation

Authorisation Flow



Implementation

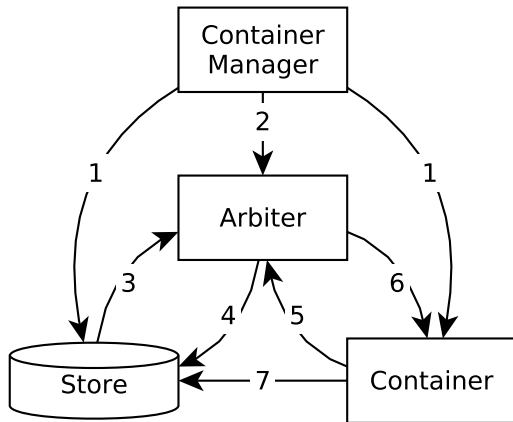
Authorisation Flow



1. CM passes unique tokens

Implementation

Authorisation Flow

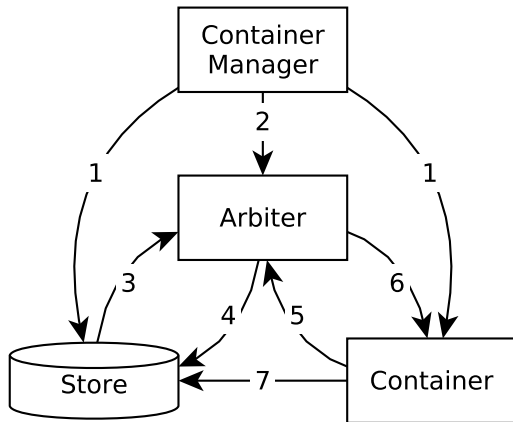


1. CM passes unique tokens

2. CM updates permissions

Implementation

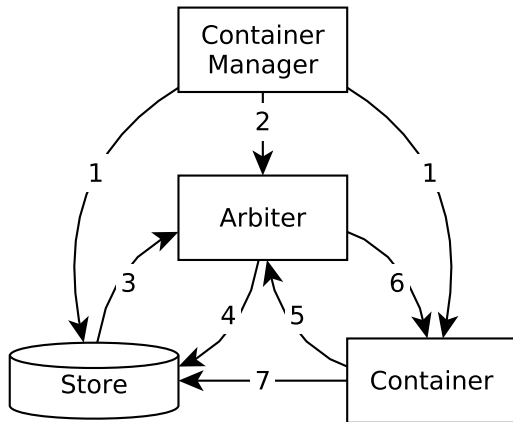
Authorisation Flow



1. CM passes unique tokens
2. CM updates permissions
3. Store registers itself

Implementation

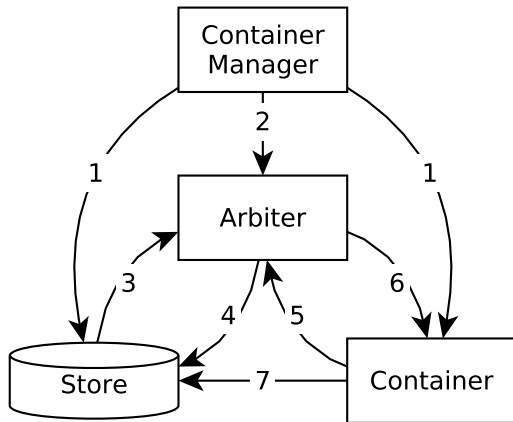
Authorisation Flow



1. CM passes unique tokens
2. CM updates permissions
3. Store registers itself
4. Arbiter responds with shared secret

Implementation

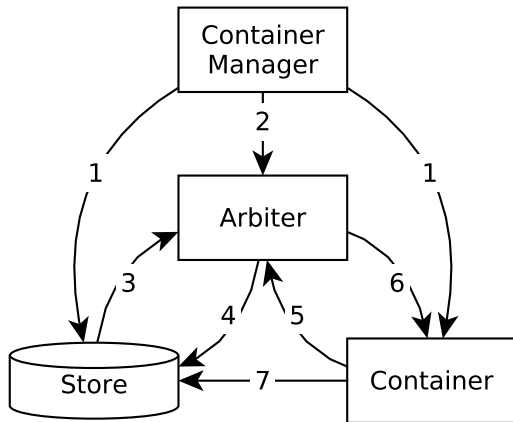
Authorisation Flow



1. CM passes unique tokens
2. CM updates permissions
3. Store registers itself
4. Arbiter responds with shared secret
5. Container requests bearer token

Implementation

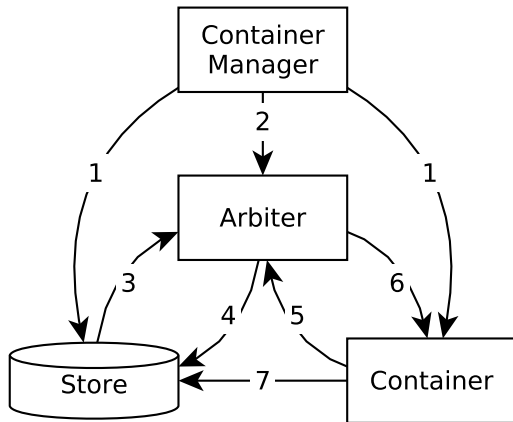
Authorisation Flow



1. CM passes unique tokens
2. CM updates permissions
3. Store registers itself
4. Arbiter responds with shared secret
5. Container requests bearer token
6. Arbiter checks and responds

Implementation

Authorisation Flow

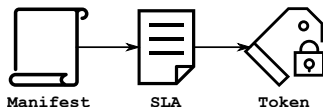


1. CM passes unique tokens
2. CM updates permissions
3. Store registers itself
4. Arbiter responds with shared secret
5. Container requests bearer token
6. Arbiter checks and responds
7. Container can now read/write to store

Implementation

Transcription of Permissions

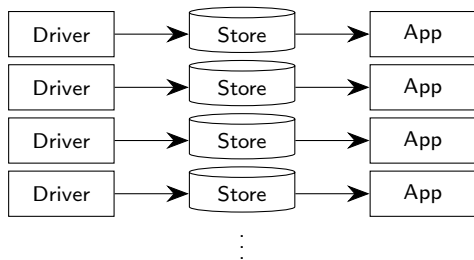
1. Drivers/apps come packaged with a *manifest*
 - ▶ Contain image metadata
 - ▶ Enumerate granular permissions for sources, concurrency, external access, and hardware
2. Users generate a Service-level Agreement (SLA)
3. The arbiter records granted permissions
4. Tokens are minted based on these



```
{  
  "name": "app",  
  "author": "amar",  
  "permissions": [  
    {  
      "source": "twitter",  
      "required": true  
    },  
    {  
      "source": "gps"  
    },  
    {},  
    {}  
  ]  
}
```

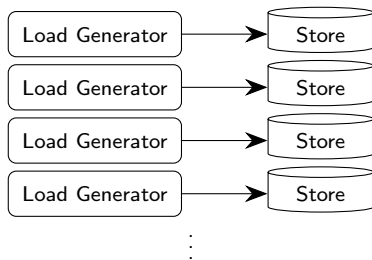
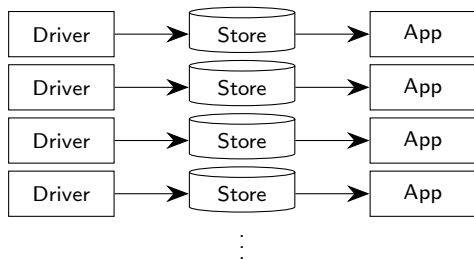
Evaluation

Scalability



Evaluation

Scalability



Evaluation

Scalability

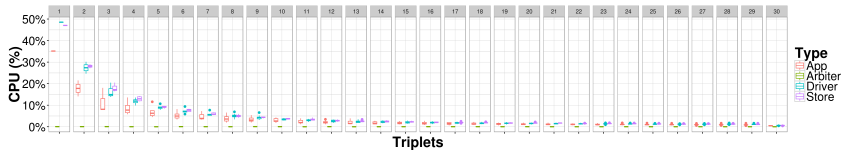


Figure: Percentage CPU Usage by Container Type

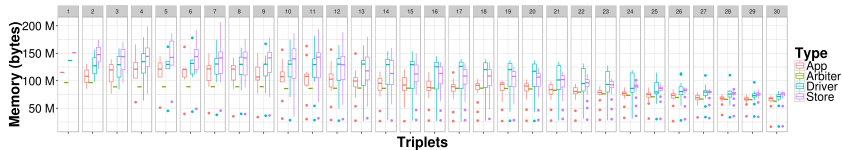


Figure: Memory Usage by Container Type

Evaluation

Scalability

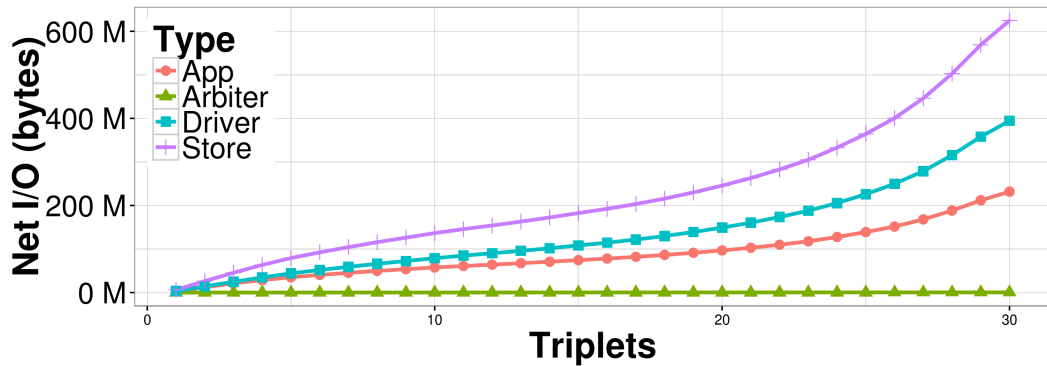


Figure: Sum Net I/O by Container Type

Evaluation

Scalability

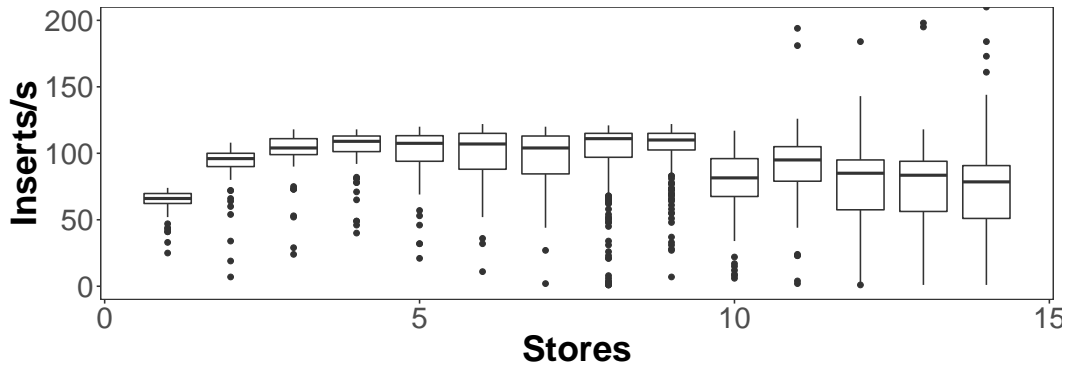


Figure: Inserts/s over Stores under Maximum Load

Evaluation

Scalability

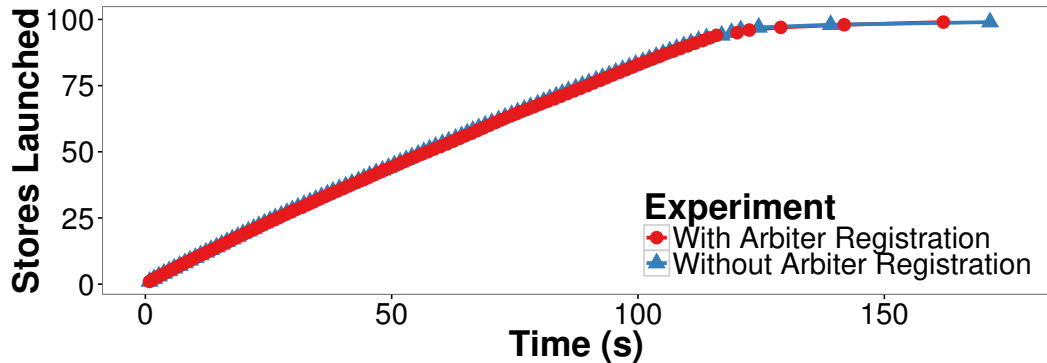


Figure: Stores Launched over Time

Evaluation

Topology

Differences in Time to Availability (TTA)

1. Device → Cloud:
65ms
2. Device → Cloud → Home:
83ms
3. Device → Home:
78ms
4. Device → Home → Cloud:
80ms

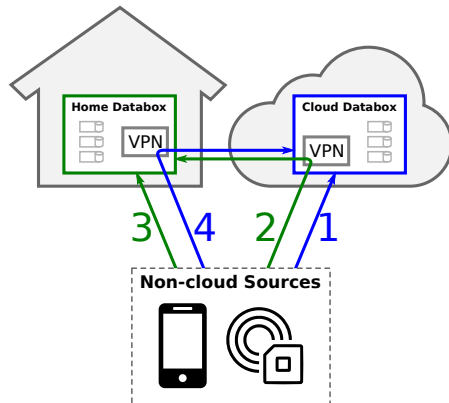


Figure: The four possible data flow scenarios tested

Evaluation

Topology

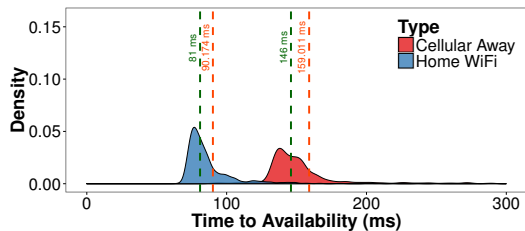


Figure: Data Time to Availability from Device to Cloud Databox Directly

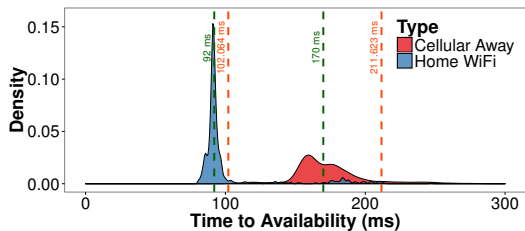


Figure: Data Time to Availability from Device to Home Databox Directly

Evaluation

Topology

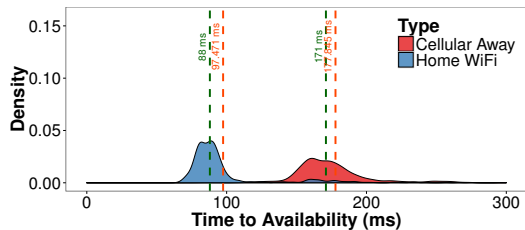


Figure: Data Time to Availability from Device to Home Databox via Cloud VPN

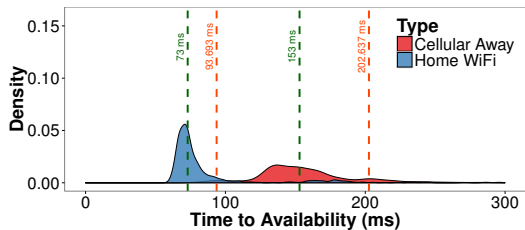


Figure: Data Time to Availability from Device to Cloud Databox via Home VPN

Evaluation

Topology

- ▶ TTA source away from home $>$ source at home
- ▶ So minor, barely indistinguishable from NTP drift
- ▶ Based on performance alone, UX indifferent
- ▶ Scenarios through home (especially when source is away) have mean shifted right due to latency spikes
- ▶ Direct connections mean lower TTA, and cloud faster than home *ceteris paribus*
- ▶ Small difference for devices as sources vs cloud servers
- ▶ For devices, processing at home $>$ in the cloud \pm NTP error even ignoring privacy advantages
- ▶ Home vs cloud — reliability vs cost
- ▶ Pure cloud only more advantageous for off-site processing (e.g. GPU-heavy image processing)

Evaluation

Time to Availability

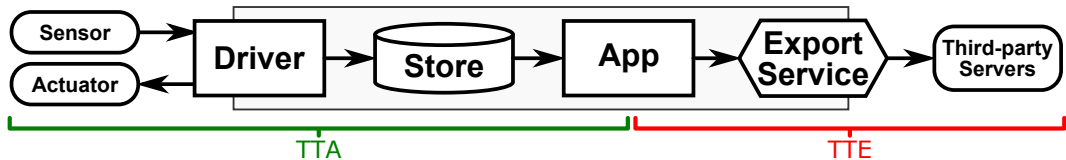


Figure: Sections of the data pipeline timed

Evaluation

Time to Availability

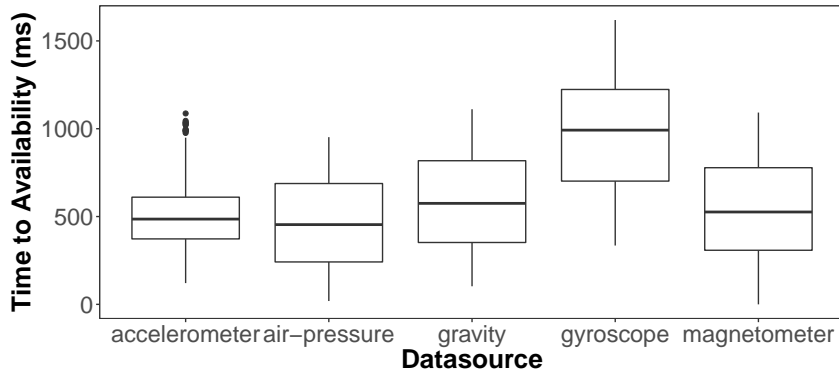


Figure: Time-to-Availability (TTA) on a Raspberry Pi for high-frequency sensors

Evaluation

Time to Availability

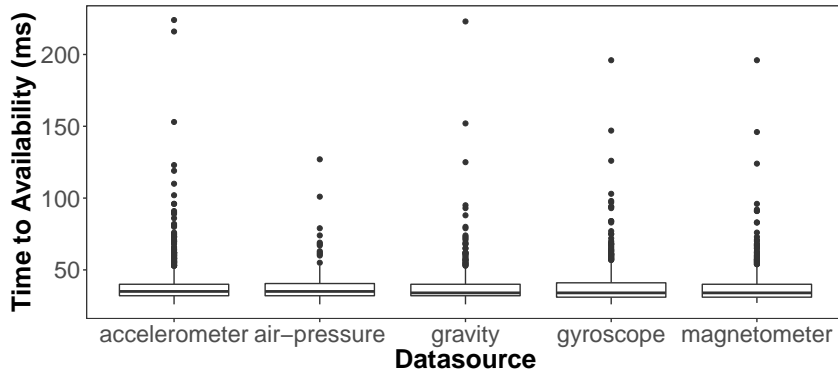


Figure: Time-to-Availability (TTA) on an Intel NUC for high-frequency sensors

The Serverless Paradigm

Background

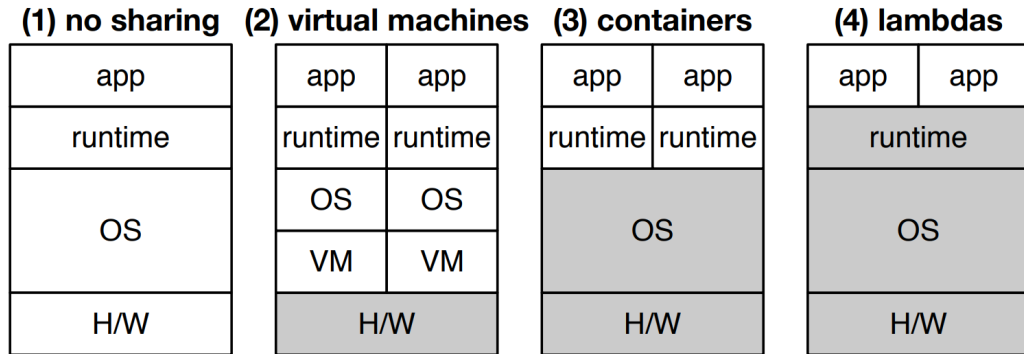


Figure 1: Evolution of Sharing. *Gray layers are shared.*

Figure: Hendrickson, et al. "Serverless computation with openlambda." Elastic 60 (2016): 80.

The Serverless Paradigm Architecture

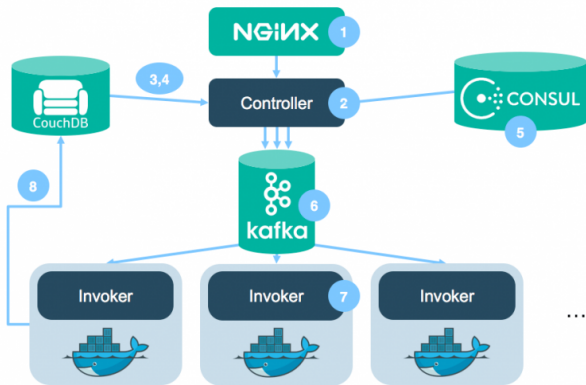


Figure: IBM's High-level OpenWhisk Architecture Diagram

The Serverless Paradigm

Architecture

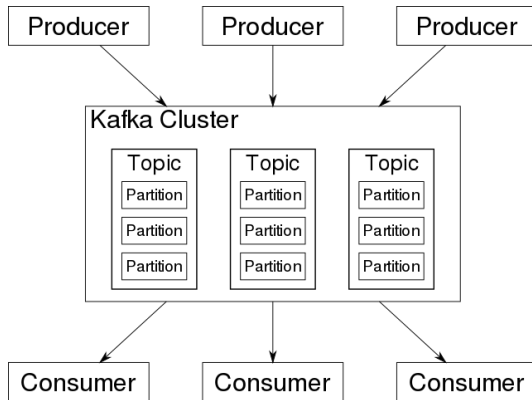


Figure: Apache Kafka High-level Architecture Diagram

Low-latency Serverless Approach

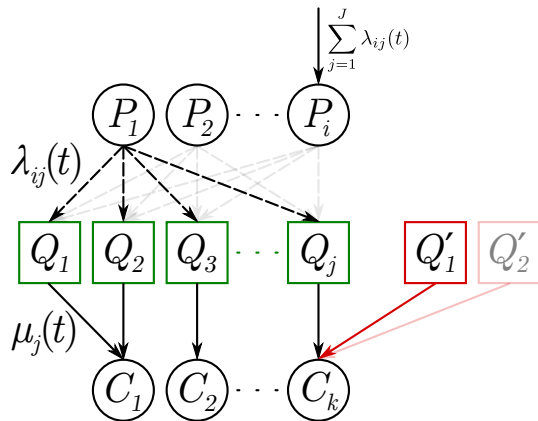


Figure: An Overview of Inter-component Relationships

Low-latency Serverless Approach

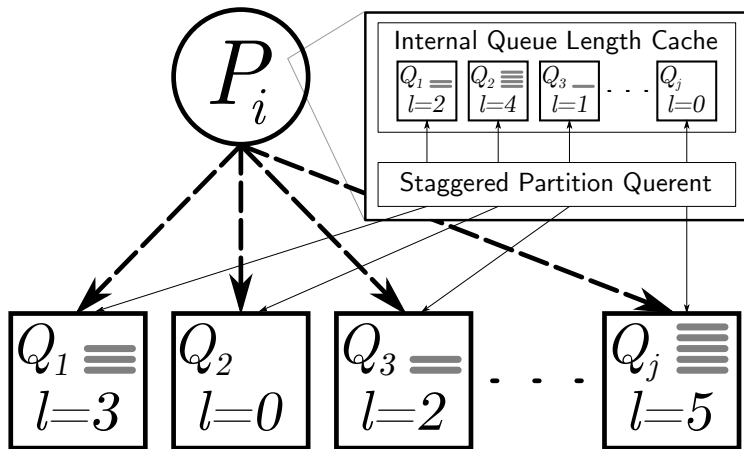


Figure: The Internal Components of a Producer

Low-latency Serverless Approach

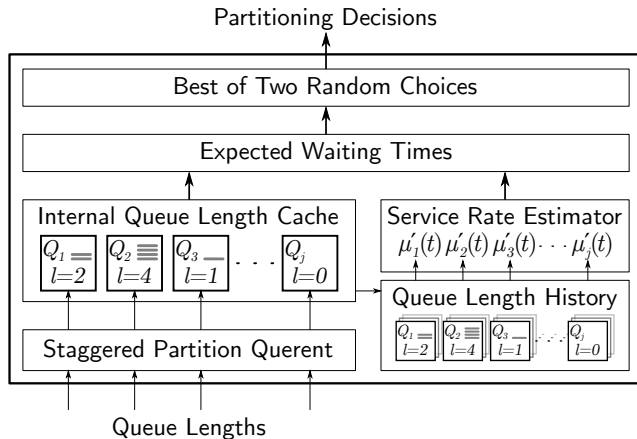


Figure: Producer-intrinsic Steps for Computing Partitioning Decisions from Stale Queue Lengths

Low-latency Serverless Simulations

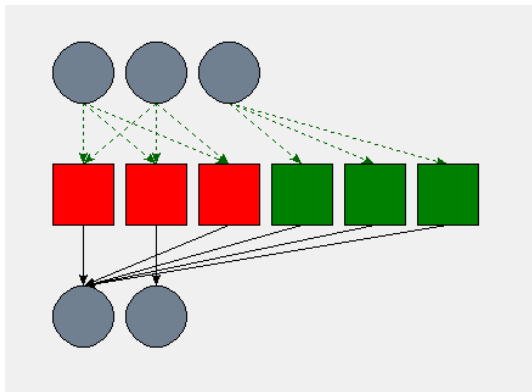


Figure: An Example of Simulation Topology

Low-latency Serverless Simulations

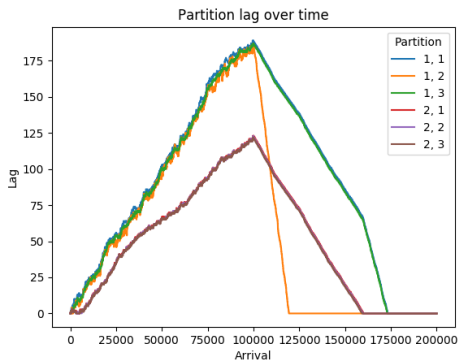


Figure: Simulation Results with Different Partitioning Algorithms

Next Steps

Privacy and Risk Metrics

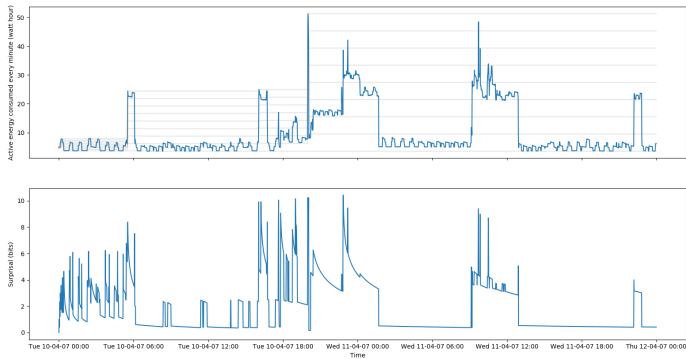
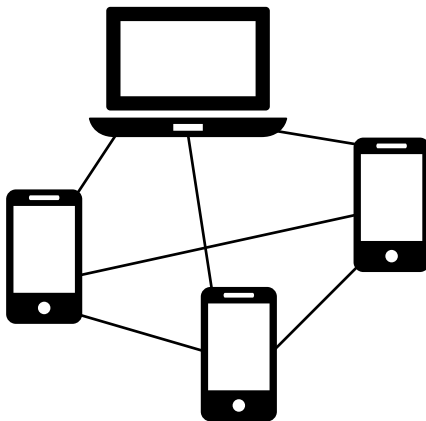


Figure: One Proof of Concept Experiment – Surprisal over Real Smart Meter Data

Next Steps

Serverless over Transient Clouds

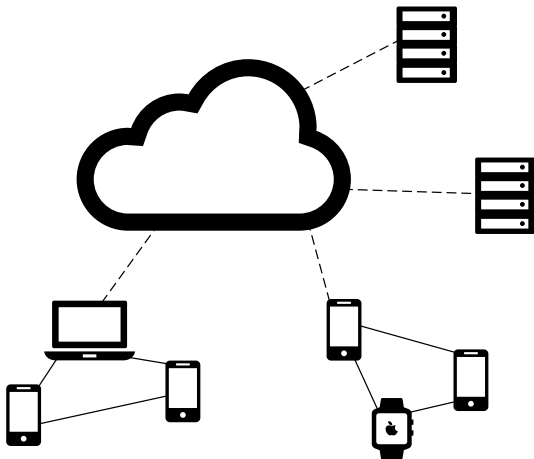
- ▶ Serverless on the edge
- ▶ Optimising for context through latency
- ▶ Processor selection based on arbitrary metrics, e.g. surprisal



Next Steps

Transient Privacy-Aware Clouds

- ▶ Encoding user-defined thresholds into bearer tokens
- ▶ Joint context at hierarchical levels
- ▶ TCACs → TPACs?



Next Steps

The Big Picture

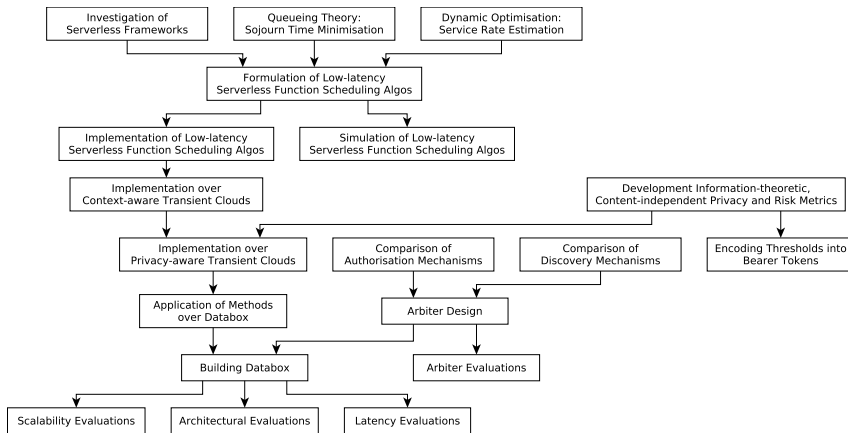


Figure: A High-level Dependency Graph of Research Activities

Thank you for your attention!

Questions?

More info: <http://yousefamar.com/>