Route-based Authorization and Discovery for Personal Data

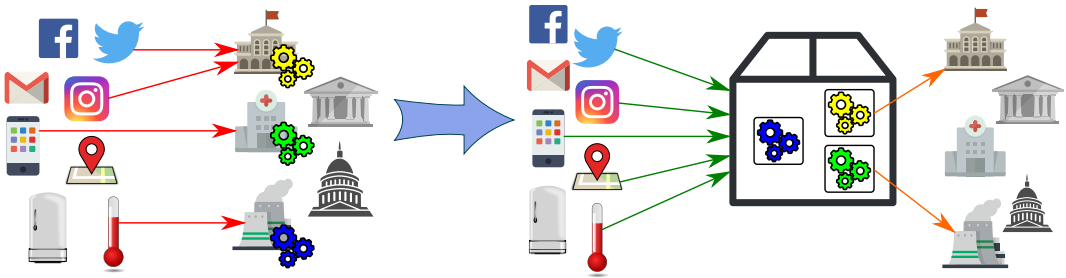
EuroDW 2017

Yousef Amar

2017-04-23

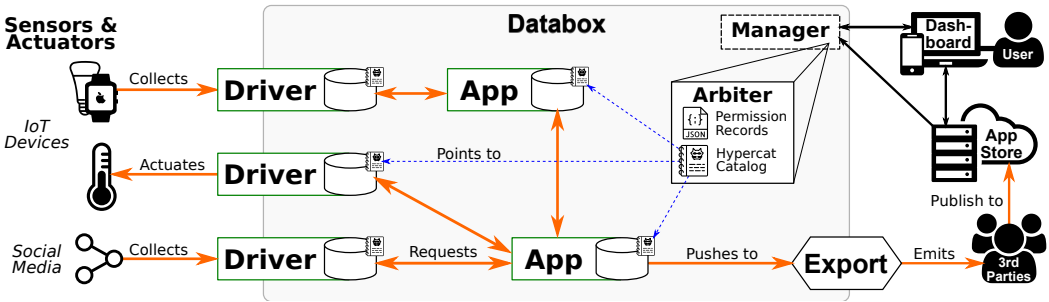
Research Context

The Databox Platform



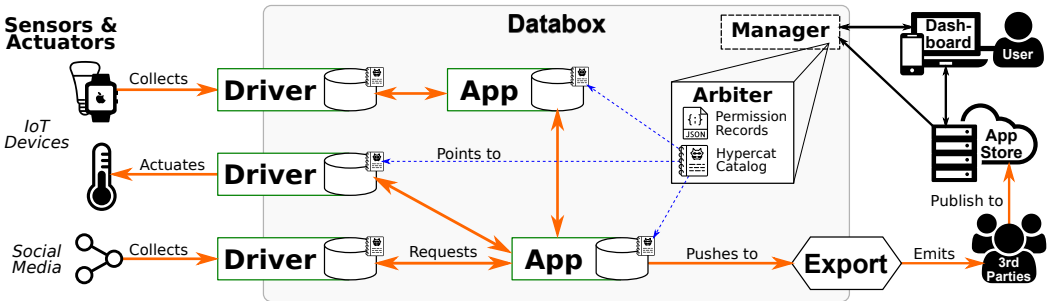
Research Context

The Databox Platform



Research Context

The Databox Platform



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

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 pervalent
- Different formats and applications — XML for REST, SOAP, OpenID

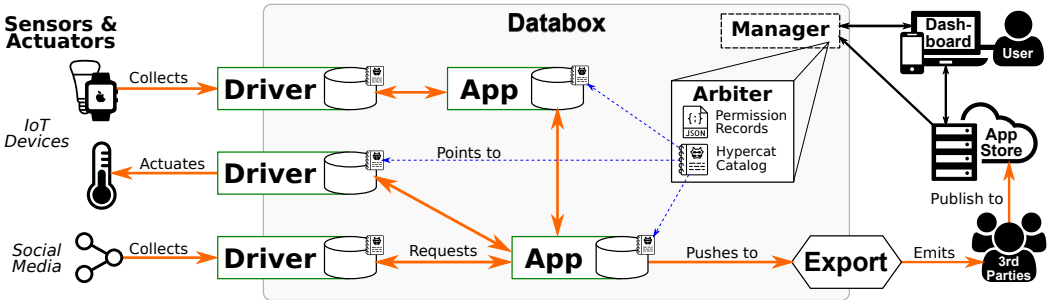
Implementation
Resource Discovery

- 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
    }]
  }]
}
```

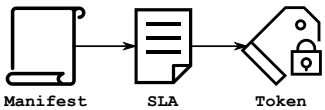
Implementation

The Arbiter



Implementation
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

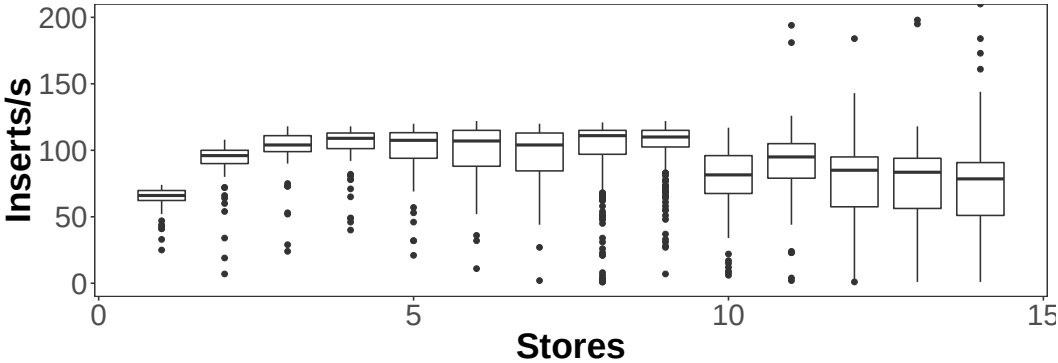


Figure: Inserts/s over Stores under Maximum Load

Evaluation
Scalability

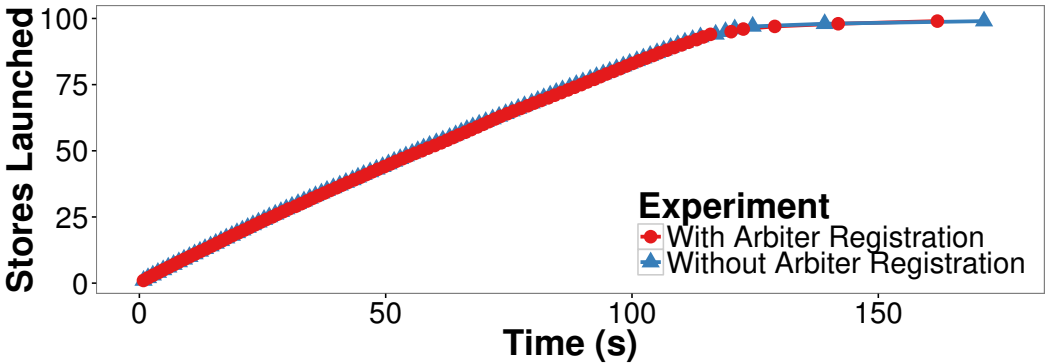
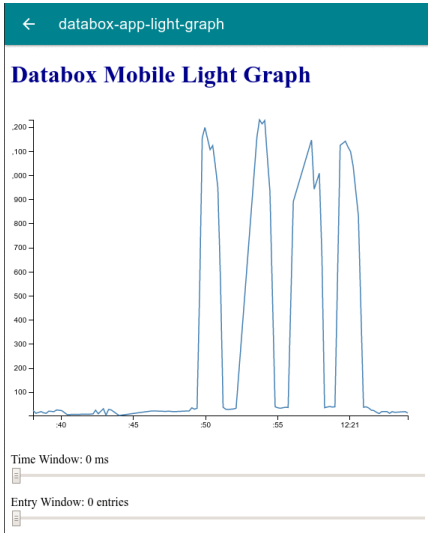


Figure: Stores Launched over Time

Next Steps

- Arbiter token minting under load evaluation
- Performance vs security when modifying token expiry
- Many areas to research, e.g. watermarking
- Many example apps and drivers, with multipurpose datavis and transformation



Thank you for your attention!

Questions?

More info: `http://www.databoxproject.uk/`
Contribute: `https://github.com/me-box`