A DIY Data Lake
Hi,
I'm Christophe.
pullreview.com

euranova.eu
ibakesoftware.com

_toch

toch
A DIY Data Lake Oasis 🌴
Where to put my data?
Why DIY?
What DIY?
Simple = Core functions
Simple = As much as possible Serverless

Building Blocks = Services, not Servers

Data Lake
Minimum Development

Focus on the API
A vernacular API: {REST}

{  
    "format": "csv",
    "name": "expenses-2016.csv",
    "source": "manual",
    "Year": "2016"
}
Straightforward Protection

```
Local

SSL

Https
```
The core of a Data Lake: A web API
The core of a Data Lake: Ingest

- How?
  - Together?
  - Meta-Data 1st?
  - Data 1st?

- Important Principles:
  - Data is identified by metadata
  - Possible delegation of down and up-load

- → Ingest in two steps:
  - POST /data + {JSON} → 202 + Data ID
  - PUT /data/ID
The core of a Data Lake: Search

- **How?**
  - Any metadata
  - HTTP Query
  - AND Based

- **Required vs Optional metadata, e.g. format**

- **Efficient search?**

- → GET /data?format=csv&year=2016 → JSON Array
The core of a Data Lake: Retrieve

- How?
  - ID!
- Possible delegation of download
- \( \rightarrow \text{GET } /\text{data}/\text{ID} \)
The core of a Data Lake: A web API

- Serverless XP
- KISS
- → web framework
  - Web API
  - Easy to deploy
- → Ruby Hanami
  - Simple, light, and fast
  - Ruby
  - I know it!
Where to ... deploy the web API?

- Any PaaS
- Criteria, e.g. automated scaling, hosted, free tier, European data center
- → Heroku
Where to ... store the metadata?

- Needs
  - {JSON} search
  - Document based storage
  - Possibly structured
  - Easy integration with Heroku
  - Common
  - Free tier

- Options: Pgsql, MongoDB, ElasticSearch

- → PostgreSQL
Where to ... store the data?

- **Needs:**
  - A blob storage
  - REST API (possible delegation)
  - European Datacenter
  - Possibly S3 compatible
  - Simple AutZ
  - Free Tier

- **Options: Heroku Addons or External Service**

- **→ AWS S3**
  - GET /data/ID → 307 redirecting to AWS S3 on GET presigned URL
  - PUT /data/ID → 307 redirecting to AWS S3 on PUT presigned URL
Simple Way to protect

- Queries encrypted through HTTPS
- Token based Authentication
- European location
- Local SSL Encryption
Demo