The Backend

Or

The embarrassing bit that no one likes to talk about

(I’m Laurie Cooper)

DigitalLabs@MMU

- We belong to MMU
- We live in the Shed
- We build tech
- We write software

DigitalLabs@MMU

- We’re an Enterprise Unit
- We model a small business
- We try to stay afloat
- We’re staffed by real developers
- We share our experiences

Today

Image credit: Tory Townsend

Today

…is all about online resources available to an application – backend:

- What’s a backend?
- What’s an API?
- What’s a data model?
- What’s REST?
- What’s an IDL?
- Why do I need security?
- What is PaaS?
- Show and Tell: RESTlet

What is a Backend?

Image credit: Tory Townsend

What is a Backend?

- Remote
  - Access via a web service
- Storage
  - Create
  - Read
  - Update
  - Delete
- Non-storage
  - Algorithms

Image Credit: Anna Grist

What is a Backend?

- Sometimes, you just need somewhere to keep your data. That’s a drive.
- Sometimes, you’ll want to do clever things with your data; search it, add more, look for correlations. That’s a database.
- Sometimes, you’ll want to access your data using an app. You’d deploy a remote database with an API. That’s a backend.
- Sometimes, you’ll want to give other developers access to your data so they can do other things with it. That’s a service!

What's an API?

Image credit: Tory Townsend

What is an API?

- Application Programmer's Interface
- An interface is:
  - A mediator between a system and its client
  - Hides implementation and complexity
  - Presents a known set of operations
- The Application Programmer is:
  - The person programming the application which will be a client to this backend

What is an API?

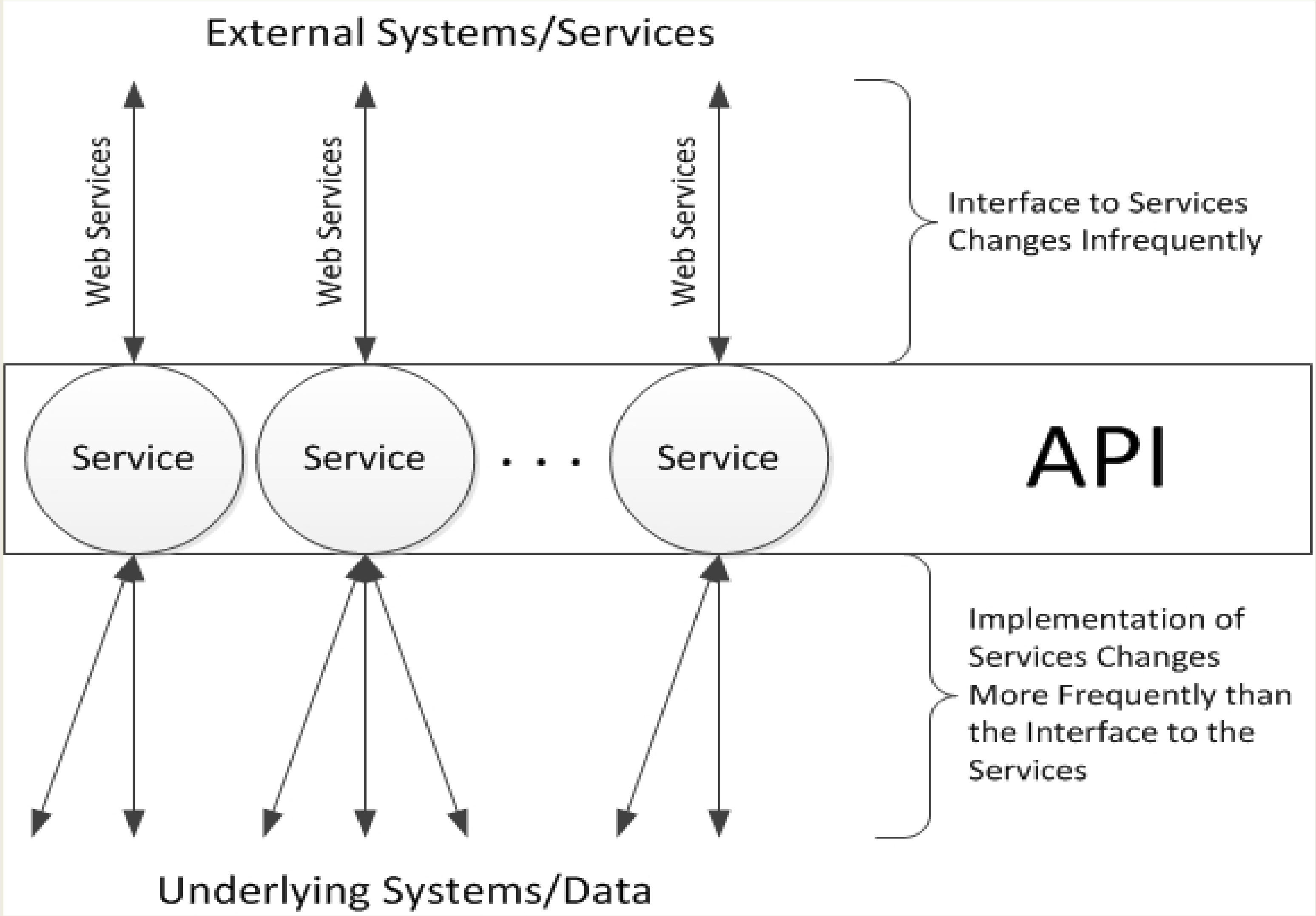

External Systems/Services
Web Services
Web Services
Web Services
Interface to Services Changes Infrequently
Service
Service
. . .
Service
API
Implementation of Services Changes More Frequently than the Interface to the Services
Underlying Systems/Data

Image credit: Roger S. Pressman

What is an API?

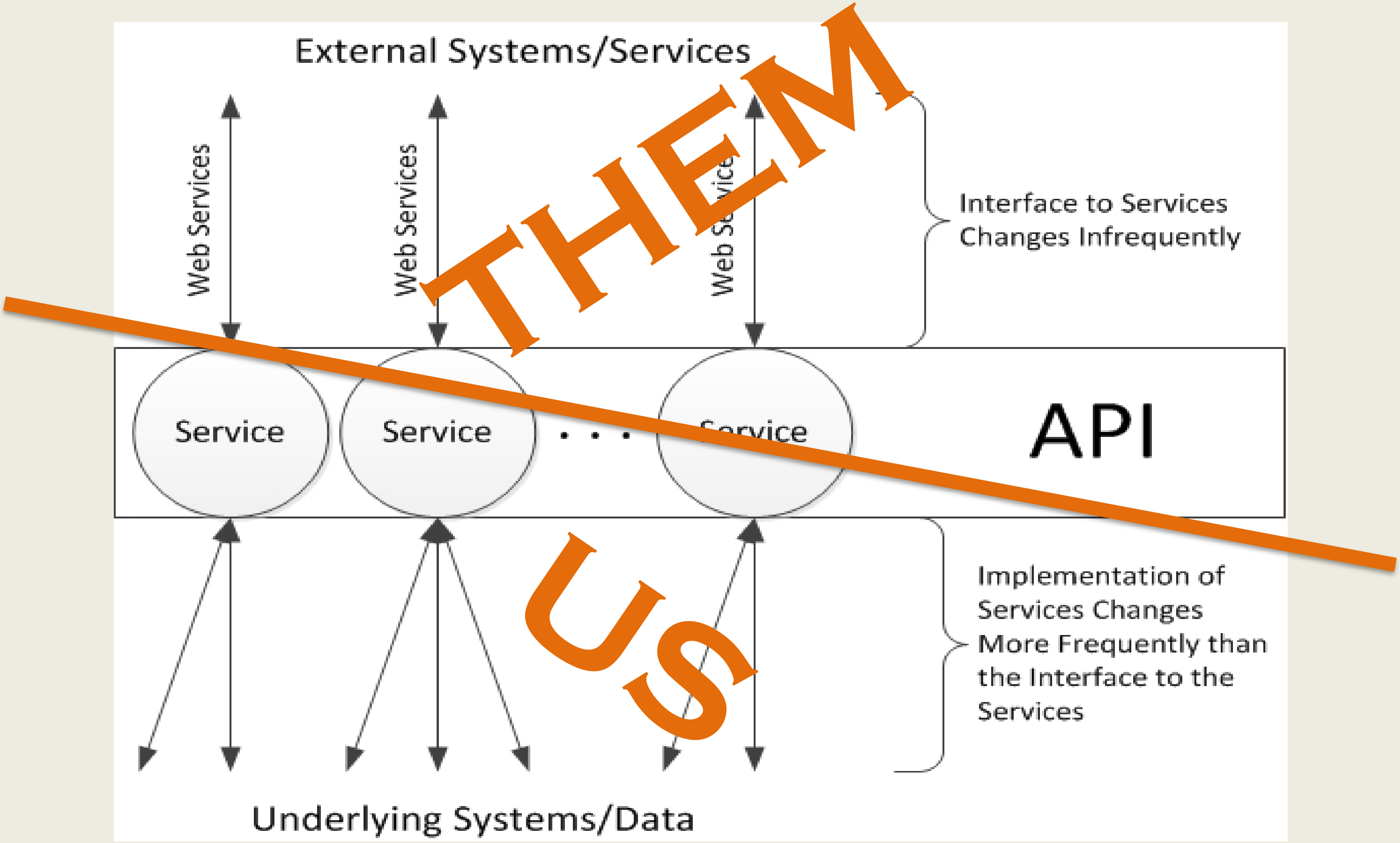

External Systems/Services
Web Services
Web Services
Web Services
Interface to Services Changes Infrequently
Service
Service
. . .
Service
API
Implementation of Services Changes More Frequently than the Interface to the Services
Underlying Systems/Data
THEM
US

Image credit: Roger S. Pressman

What is an API?

A backend API runs on a server:

- Sits in front of your resource
    - Protects it
    - Simplifies access to it

What is an API?

You create the API:

- Define the operations it exposes
- Define the data it returns

What is an API?

You create the service which implements it:

- Wait for requests from clients
- Fetch database data
- Fetch data from other services
- Return data

What is an API?

Clients communicate with a service via a protocol:

- REST (Representational State Transfer)
- SOAP (Simple Object Access Request Protocol)
- Lots of Others

What is an API?

Pass data over the API as a payload of the protocol:

- JSON
- XML
- Lots of Others

(you can even use CSV)

What makes a ‘good’ API?

Image credit: Tory Townsend

What makes a good API?

- Efficient
  - Minimise calls
  - Minimise bandwidth
- Scalable
  - Larger datasets
  - Many users
- Well documented
  - Essential
  - Accurate
- Easy to use
  - KISS *

*Keep It Simple Stupid. Attributed to Kelly Johnson, Lead Engineer at Lockheed Martin's Skunk Works

What makes a good API?

APIs are *designed*:

- Operations
- Data model

What is a Data Model?

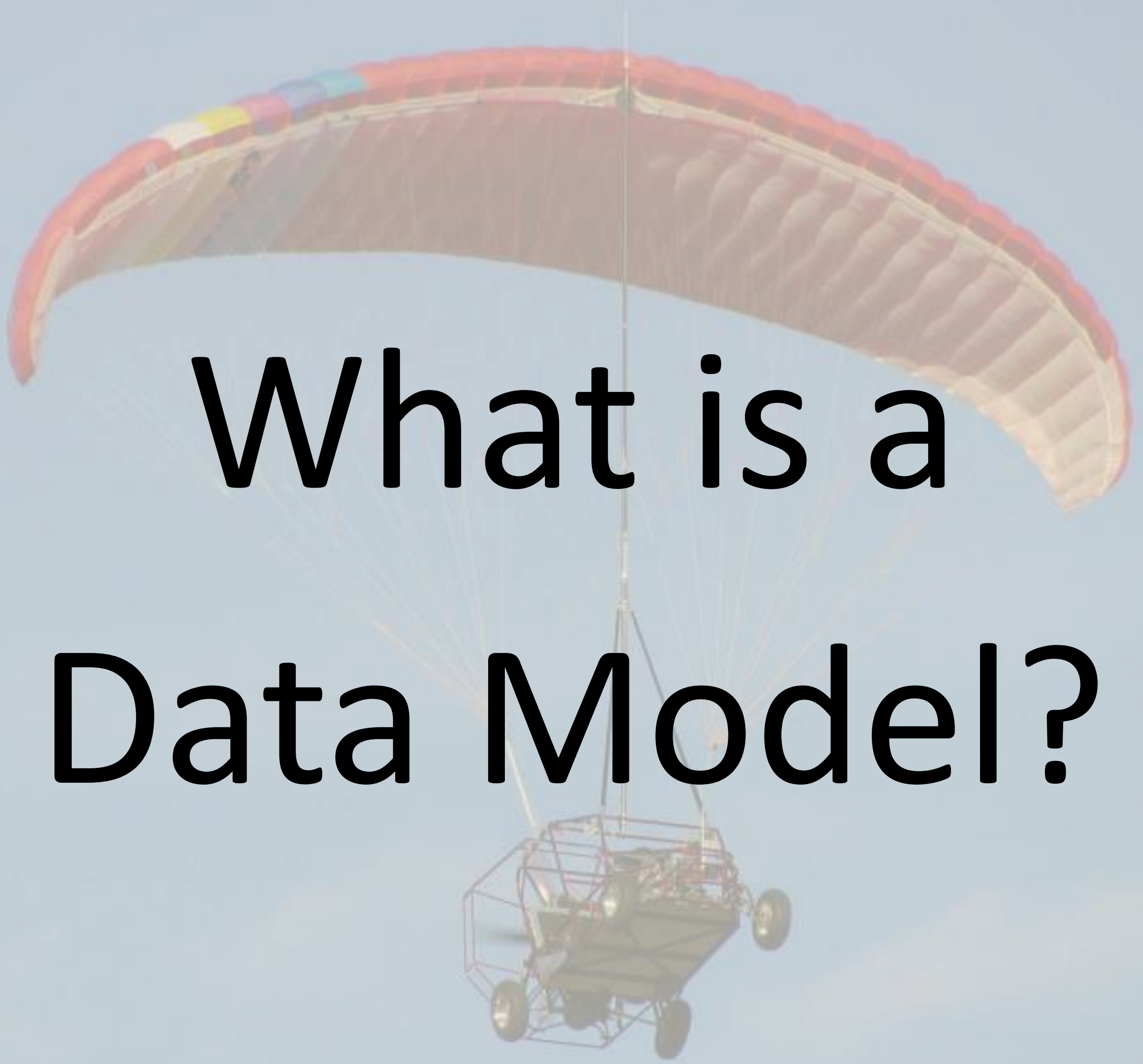


Image credit: Tory Townsend

What is a data model?

A data model:

- Represents the parameters of the operations exposed by the API
- Same or subset of your underlying database
- … but doesn’t have to be ☺
- Is ‘typed’

What’s REST?

Image credit: Tory Townsend

What is REST?

REST (Representational State Transfer)

- set of architectural principles
- Sits on / developed in parallel with HTTP 1.1 - Rich
- In 2000 by Roy Feilding
- Changes the state of the 'application' using links and operations

What is REST?

REST (Representational State Transfer)

- A URL points to a resource
- Collection:
  - www.cars.com/registrations/
- Item:
  - www.cars.com/registrations/AB45PRZ/

What is REST?

RESTful (web services)

- HTTP used to interact with application’s resources
- Request methods used to do this:
  - PUT – creates
  - GET (safe method)
  - POST – updates completely
  - DELETE - removes
  - PATCH – changes
- Standard HTTP responses returned
- ‘Stateless’ protocol - no concept of a session

What is REST?

RESTful (web services)

The MIME type tells the client what to expect. Often:

- application/json
- application/xml

What is REST?

RESTful (web services)

Send a GET request:

- http://www.cars.com/registrations?page=1&size=2

Receive a response:

- Response Code: 200
- … and…

What is REST?

RESTful (web services)

GET http://www.cars.com/registrations?page=1&size=2

```
Response Header:
{
  "content-type": "application/json"
}
Response Body:
[
  {
    "id": "a21b6727-9c66-48fd-a98d-d2c809645929",
    "registration": "R1 LEY"
  },
  {
    "id": "5e599cad-c4e7-489e-8060-cf205ebe2d49",
    "registration": "CAR 70Y"
  }
]
```

What is REST?

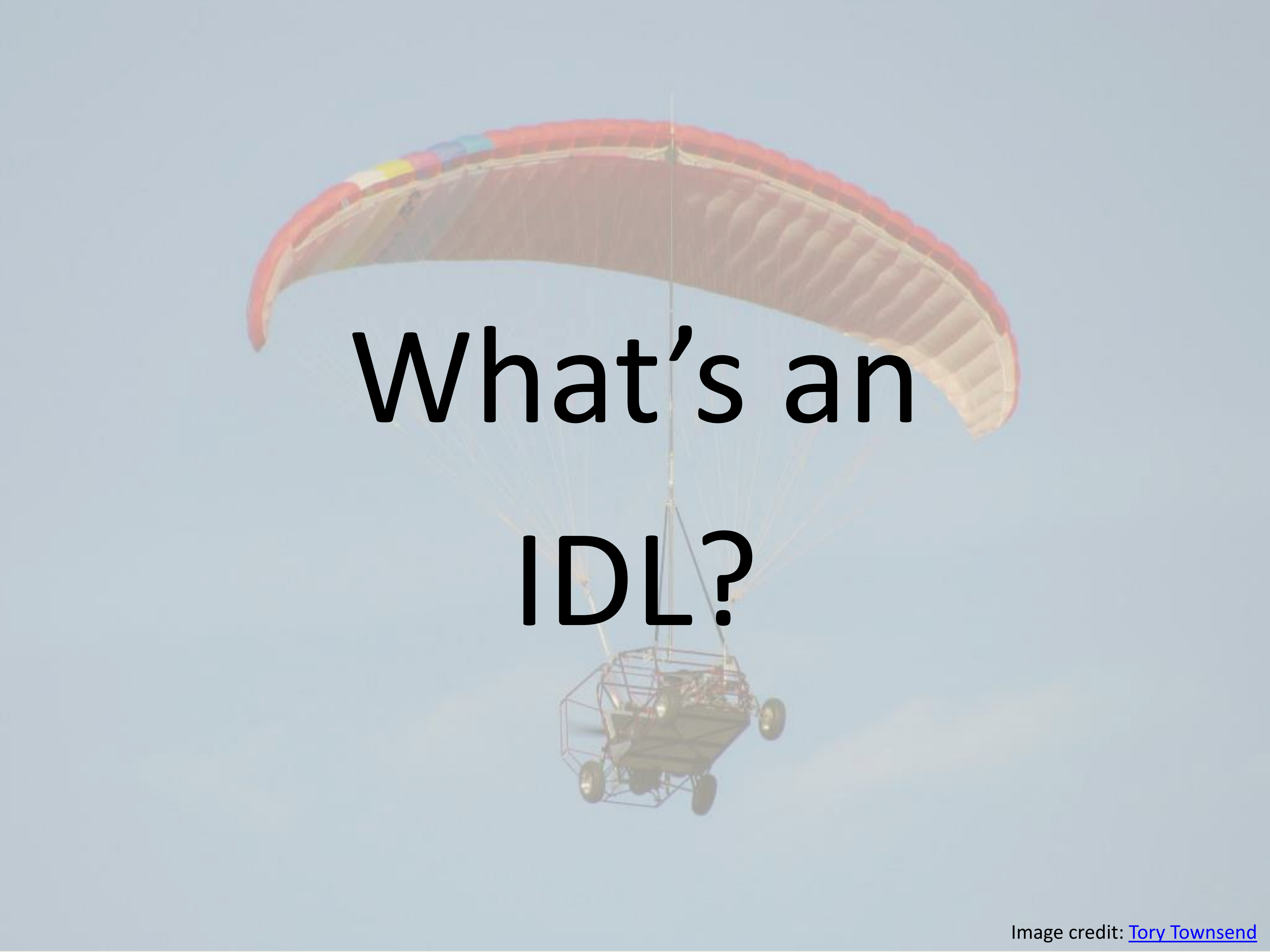
There are lots of service technologies.
Choosing one is a compromise.
We use REST, because we need to get things done:

- Simple, yet rich
- Everywhere
- Well-tooled

What is REST?

There are lots of service technologies.
Choosing one is a compromise.
We use REST, because we need to get things done:

- Not fast, not slow
- Not too verbose
- Easily debugged



What’s an IDL?

Image credit: Tory Townsend

What’s an IDL?

Interface Definition Language

- SOAP -> WSDL
- REST -> OAS (swagger)
- gRPC -> Protocol Buffers
- Lots of others!

What’s an IDL?

Interface Definition Language

- Formal definition of an interface
  - Operations
  - Data
- Enables Code Generation from an API ☺

What’s an IDL?

Interface Definition Language

- For REST there are lots of tools:
  - swagger.io
  - RESTlet studio

These will export code from an IDL. Create a Java EE / NodeJS server from an interface definition.

What’s an IDL?

Interface Definition Language

Alternative: create an entire server from your database

- RESTlet cloud
- PostgREST

“It’s so fast to develop, it feels like cheating!”

Why do I need Security?

Image credit: Tory Townsend

Why do I need security?

- If you deal with any kind of user data, then you need to be aware of:

  - The Information Commissioner’s Office (ICO)
  - The Data Protection Act
  - The General Data Protection Regulations

- If you don’t need to, don’t store personally attributable / identifiable information AT ALL. EVER.

Why do I need security?

Securing websites and databases is tricky.

- It is an arms race.
- Use Platforms As A Service to delegate:
  - Complex configuration
  - User Authentication

What’s a PaaS?

Image credit: Tory Townsend

What’s a PaaS?

Platform as a Service

A server system which provides a set of services, accessible online.

In particular for us:

- Database Hosting (Mongo, MySQL, Postgres…)
- API services hosting
- User accounts and authentication

What’s a PaaS?

We use Heroku as our easy-to-use service provider.

Why?

- Much of the configuration previously seen with webservers is hidden.
- That means it’s more difficult to make mistakes

What’s a PaaS?

- We use Auth0 as our easy-to-use authentication provider.

Why?

- We can delegate all user accounts to experts
- They provide a robust method by which we can authenticate users.
- That means it’s more difficult to make mistakes

What’s a PaaS?

- We use GitHub Pages as our easy-to-use front-end provider.

Why?

- It’s easy.
- We can host static web pages which authenticate through a third-party, and then access data via an API.
- Pushed immediately from our GitHub repo

10 minute Backend

Image credit: Tory Townsend

10 minute Backend

We use RESTlet cloud to prototype our backends:

- Fast
- Free
- Fauthenticated Access

10 minute Backend

Example!

We want to catalog the wildlife in our urban areas

- Specify a Postcode and the animal, bird, fish, tree, bush we saw
- Everyone can use a website form to add a quick observation
- Everyone can use a general website to see a map

10 minute Backend

User Story!

Alice is walking a short distance from her home and sees a Jay (that’s a bird) in a Larch (that’s a tree).
Quickly, she whips out her phone, and goes to http://theurbanwild.org/
Immediately, she is presented with a ‘Log your Wild’ screen, which has input boxes for ‘Wild’ and ‘Where’, and a big, friendly ‘Log it!’ button.
In the ‘Wild’ input box, she starts to type ‘J..’ and a suggestion pops up ‘Jay’. She accepts the suggestion.
In the ‘Where’ input box, is the postcode of her current location. She accepts this by simply pressing ‘Log it!’.
The website responds with a grateful ‘Thanks! Your Jay has been logged! 15 times at your location today ’

10 minute Backend

Thing:
String name

Event:
String postcode
Timestamp date
Thing thing

… off to RESTlet

10 minute Backend

Here’s one I prepared earlier…

Got CURL? Type this:

```
curl --request GET   --url https://theurbanwild.restlet.net/v1/things/
```

Get this:

```
[
  {
    "id": "cfe54240-d07c-11e7-89e3-
c563fcbeb0a7",
    "name": "Jay"
  },
  {
    "id": "e3b48470-d07c-11e7-88f3-
c7a46f1bb72e",
    "name": "Squirrel"
  }
]
```

10 minute Backend

Now get all the events involving a Jay…

Type this:

```
curl --request GET  --url https://theurbanwild.restlet.net/v1/events/?thing=cfe54240-d07c-11e7-89e3-c563fcbeb0a7
```

Get this:

```
[
  {
    "id": "36bd3b30-d07d-11e7-88f3-c7a46f1bb72e",
    "postcode": "M16 5GD",
    "date": 1511395200000,
    "count": 3,
    "thing": "cfe54240-d07c-11e7-89e3-c563fcbeb0a7"
  },
  {
    "id": "6f9e69d0-d085-11e7-89e3-c563fcbeb0a7",
    "postcode": "M16 0BT",
    "date": 1511395200000,
    "count": 1,
    "thing": "cfe54240-d07c-11e7-89e3-c563fcbeb0a7"
  }
]
```

10 minute Backend

Now get all the events involving a Jay…

Type this:

```
curl --request GET  --url https://theurbanwild.restlet.net/v1/events/?thing=cfe54240-d07c-11e7-89e3-c563fcbeb0a7
```

Get this:

Java Epoc Time (s)

```
[
  {
    "id": "36bd3b30-d07d-11e7-88f3-c7a46f1bb72e",
    "postcode": "M16 5GD",
    "date": 1511395200000,
    "count": 3,
    "thing": "cfe54240-d07c-11e7-89e3-c563fcbeb0a7"
  },
  {
    "id": "6f9e69d0-d085-11e7-89e3-c563fcbeb0a7",
    "postcode": "M16 0BT",
    "date": 1511395200000,
    "count": 1,
    "thing": "cfe54240-d07c-11e7-89e3-c563fcbeb0a7"
  }
]
```

Wrap-up

Image credit: [Tory Townsend]

RESTlet

Pros:

- Great for prototyping / Learning
- Full API creation: API and DB
- Exports IDL
- Generates skeleton server code
- Generates skeleton client code
- Interfaces to other DB stacks
- Deploys
- Authenticates (with Auth0)
- Backups data

RESTlet

Cons:

- DB only. No extra functionality
- No location functions (‘near’ here)
- Limited (case sensitive) search
- Can’t export DB from full stack

But, much of this can be accounted for in a ‘smart’ client.

RESTlet

Advice:

- Use it
- Flexible enough to be used as a gateway to a ‘proper’ database
- Can use with other databases and even Google Sheets
- Cuts development time for basic projects from days to minutes
- You CAN learn a great deal from it.