Anatomy of a Cloud Environment

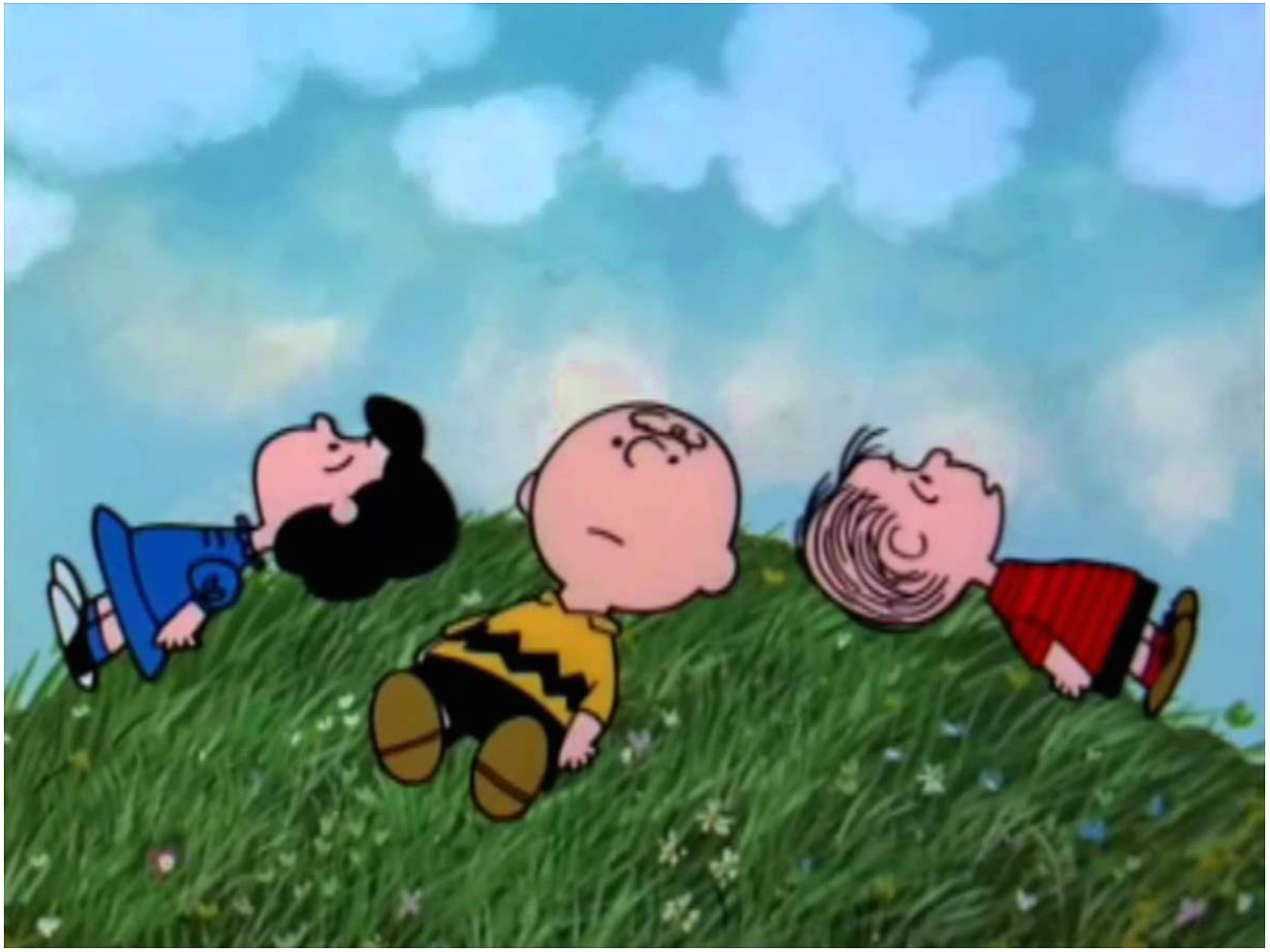
Jeffrey Bisti
IBM Poughkeepsie
jbisti@us.ibm.com

Agenda

- **Identifying Building Blocks**
- **Drawing out Implementations**
- **Planning for the Future**

Getting On The Same Page

Yes, this is still necessary



Here's what NIST says

or: why can't we just all **pretend** we're actually doing this?

Requirements:

1. **On-Demand Service** – Don't need to talk to a guy

2. **Broad Network Access** – From anywhere to anything

3. **Resource Pooling** – Abstraction of actual assets and resources to what is "given" to the consumer

4. **Rapid Elasticity** – Capabilities can be provisioned and released, possibly without any intervention

5. **Measured Service** - Resource usage can be monitored, controlled, and reported

Service Models as a Service

You get a cloud, You get a cloud, Everybody gets a cloud!!!

Software as a Service (SaaS)

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user- specific application configuration settings.

http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

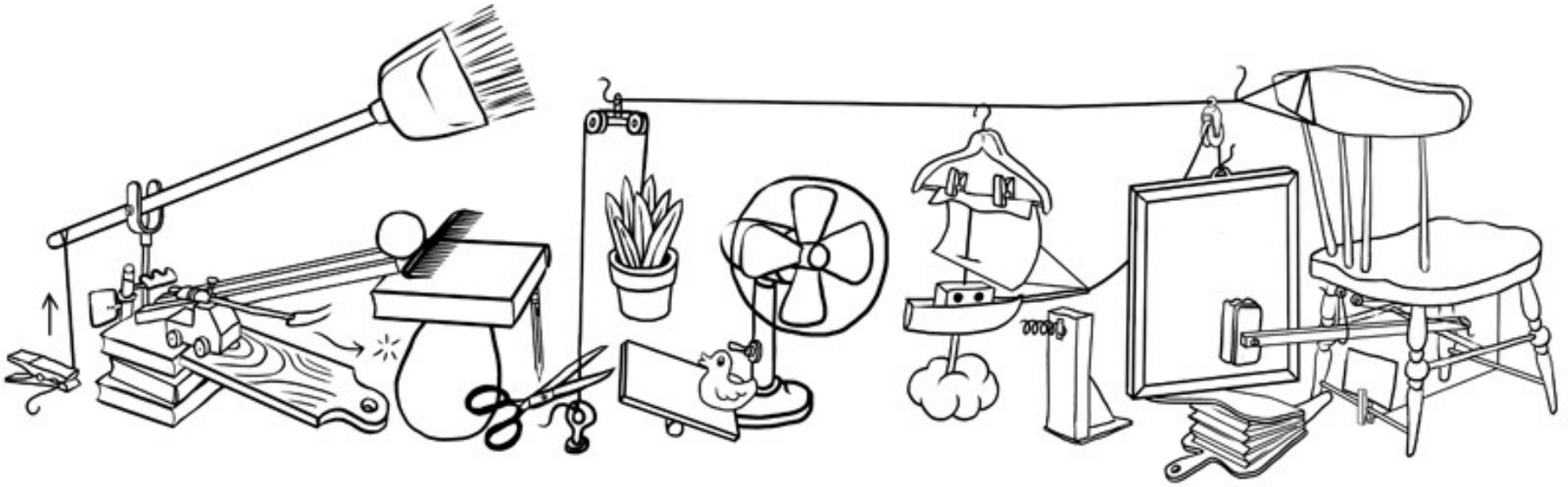
Service Instances

"So are you talking about a **SERVER** or a *server* here?"

Service	**Jeff's Location Detail Service** Input: Lat/Long Geolocation Output: Elevation and current temperature
Service Instance	Jeff's Location Detail Service, provisioned specifically for **Frank's Awesome Mobile App** **"FrankCo Weekend Hike Planner"**
System Resources	2 new userids, 50MB of storage, 4 tcpip ports… May also involve additional systems, but this is not part of the service or service instance

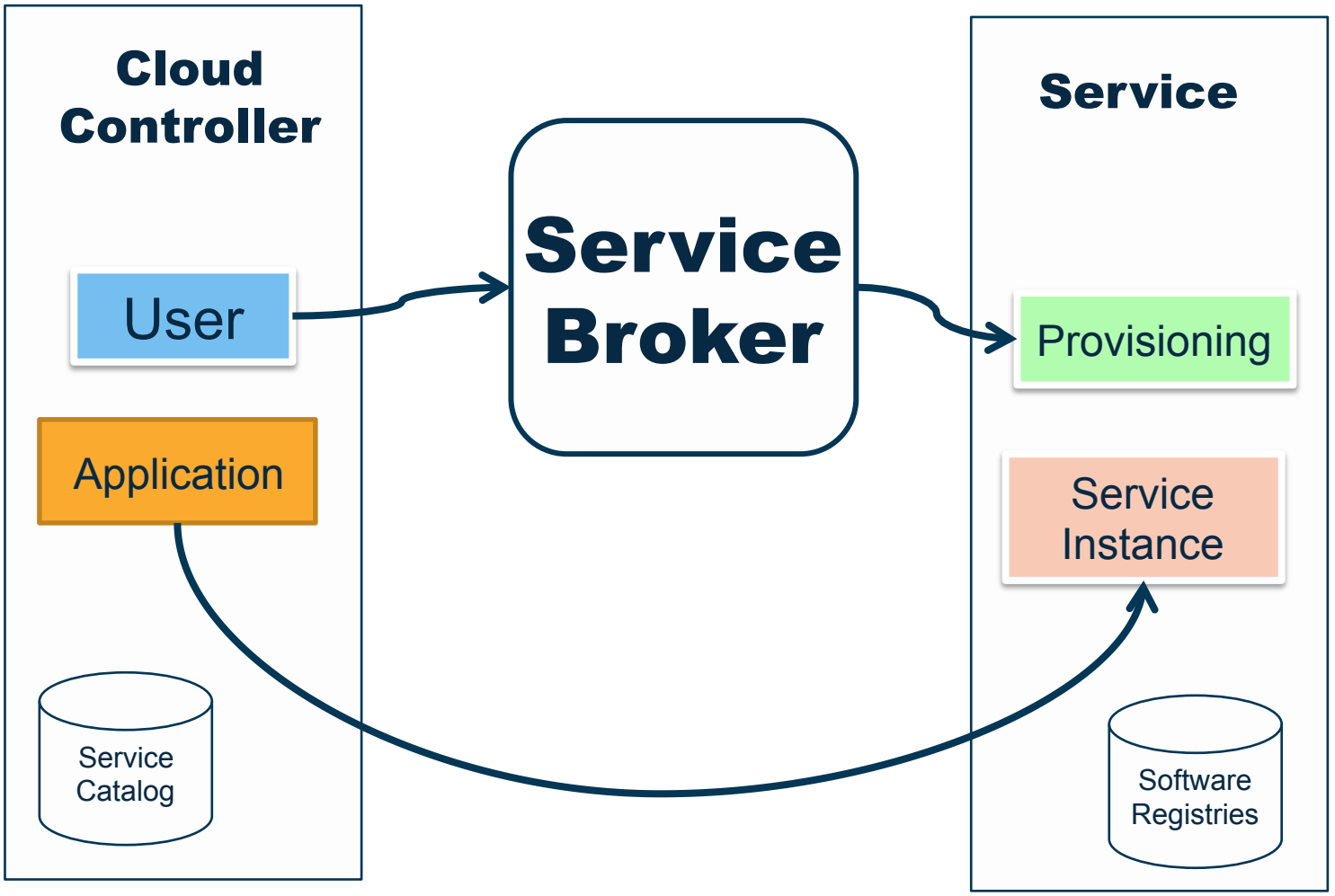
What the Developer Sees…

What the backend developer KNOWS



Service Broker:

Because Good Fences Make for Good Neighbors



Service Broker:
8 Simple Rules for Using My Service

- **Service Catalog** - Cloud Controller will initially fetch this endpoint from all brokers and make adjustments to the user-facing service catalog stored in the Cloud Controller database

- **Provisioning** – Create whatever resources are necessary to create a new service for this developer
  Deprovisioning – Destroy the resource that was created for this specific instance

- **Updating a Service Instance** – Allows users (consumers) to modify an existing service instance contract

- **Binding** – Creates and associates a credential so an individual application's access can be controlled
- **Unbinding** – Allow to revoke/cancel that application's access

- **Broker Errors** – Provide resolution between HTTP error codes and meaningful messages (ex: HTTP 414 = Attempted to decrypt string that was not encrypted)
- **Orphans** – Instances that were created, but maybe did not complete in time, or created errors for some other reason. These will be deleted.

API Management

Get started with API Management

You can design, publish, and manage APIs through the API Manager console.

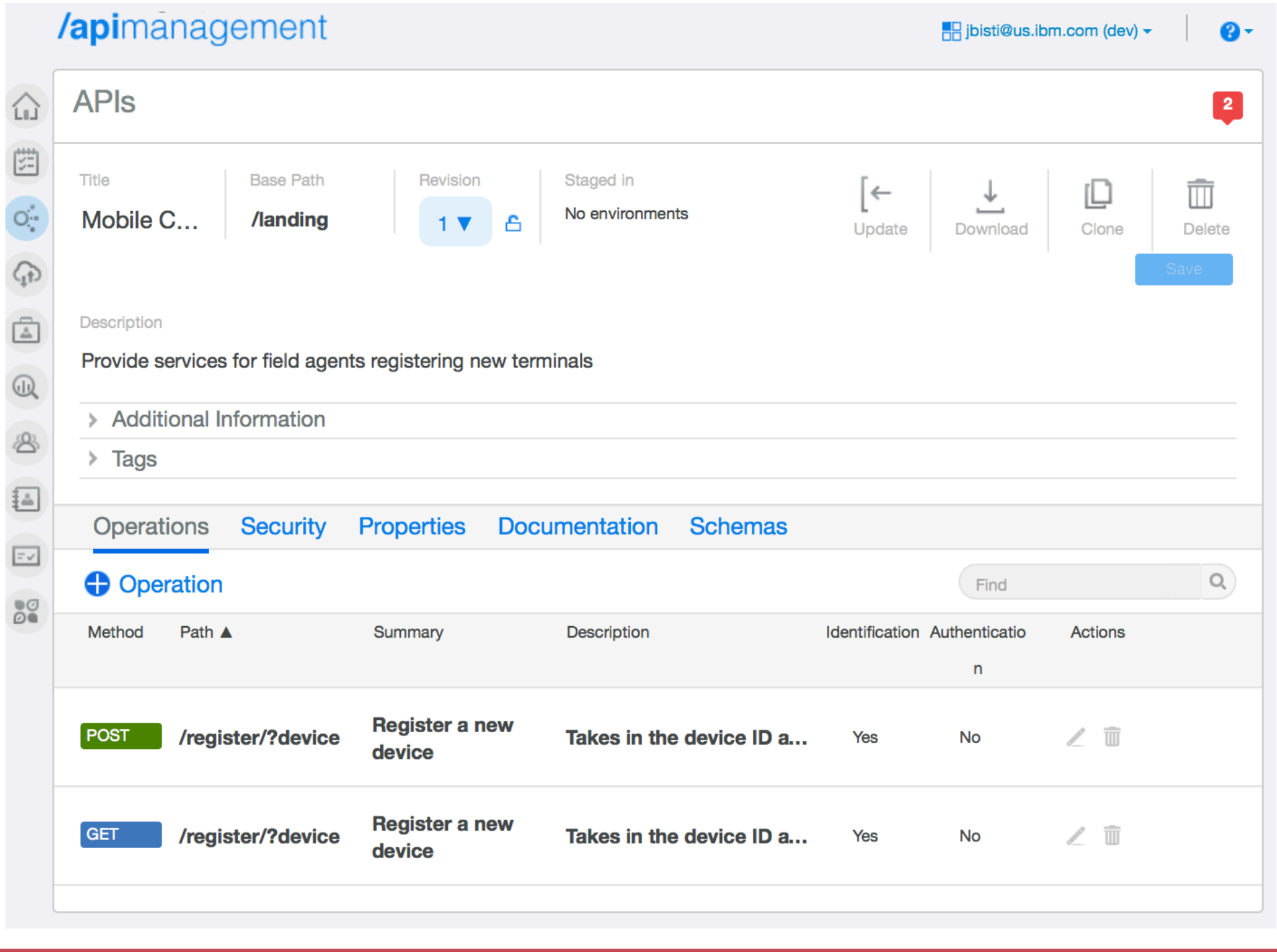
Import APIs, or compose a new one.

Create a new plan, add resources and rate limits, and deploy.

Invite other Bluemix organizations to use your APIs.

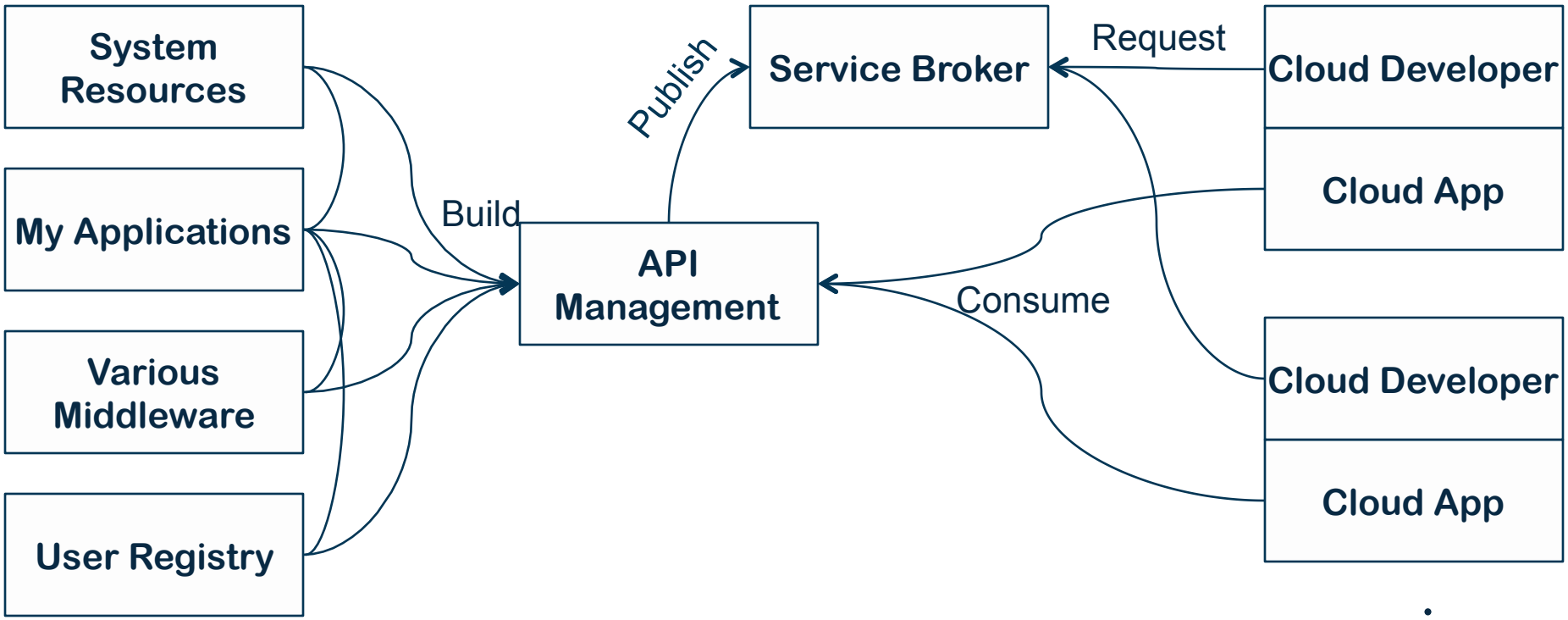
Publish your plan.

Application developers will be able to discover and consume your APIs from the Bluemix catalog.


/apimanagement
jbisti@us.ibm.com (dev)
APIs
2
Title
Mobile C...
Base Path
/landing
Revision
1
Staged in
No environments
Update
Download
Clone
Delete
Save
Description
Provide services for field agents registering new terminals
Additional Information
Tags
Operations
Security
Properties
Documentation
Schemas
Operation
Find
Method
Path
Summary
Description
Identification
Authentication
Actions
POST
/register/?device
Register a new device
Takes in the device ID a...
Yes
No
GET
/register/?device
Register a new device
Takes in the device ID a...
Yes
No

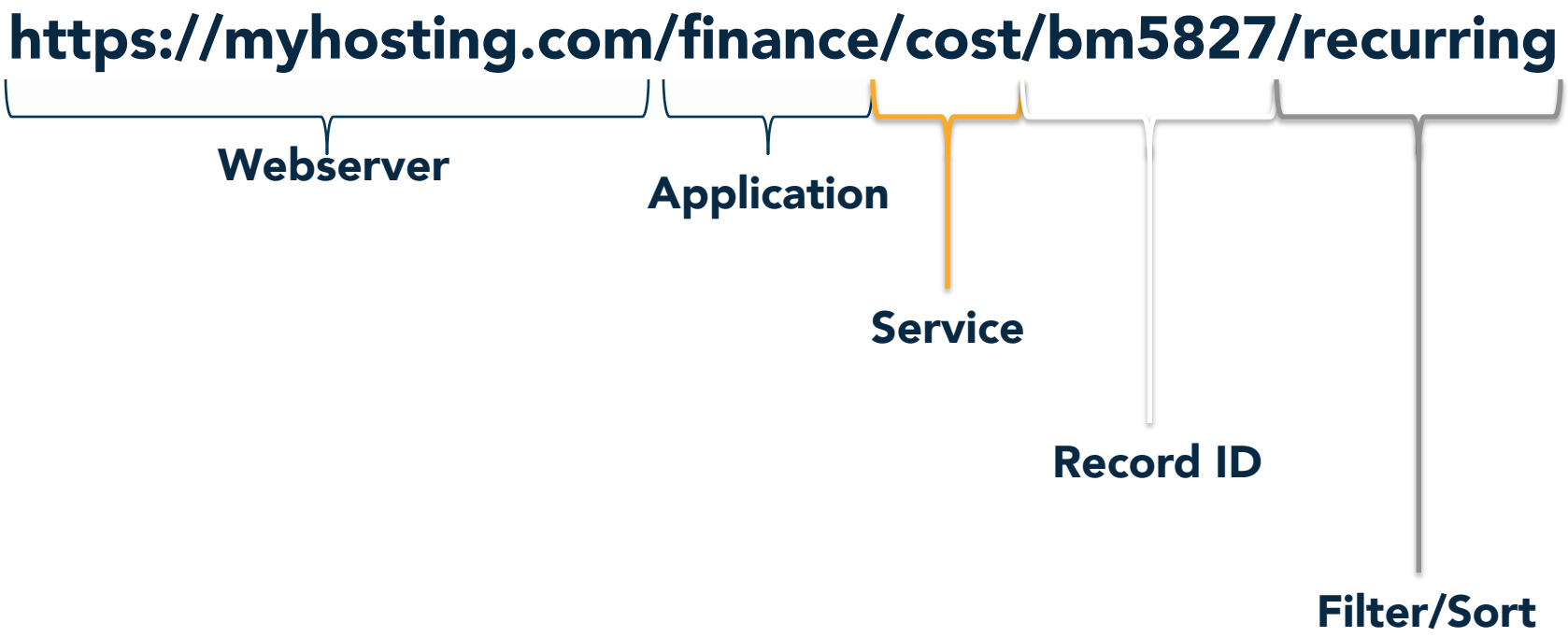
Now, in the right order…

not shown: put your left foot in, shake it all about



Anatomy of a URI

The query parameter's connected to the… endpoint



"Give me all of the recurring charges for the project filed under BM5827…"

The Microservice Model

Hooks and handles into existing infrastructure

Provide services that appeal to ground-up programmers:

Examples:

- File Store
- Crypto
- Data Transformation Facilities
- Single-service DB lookups / resolvers

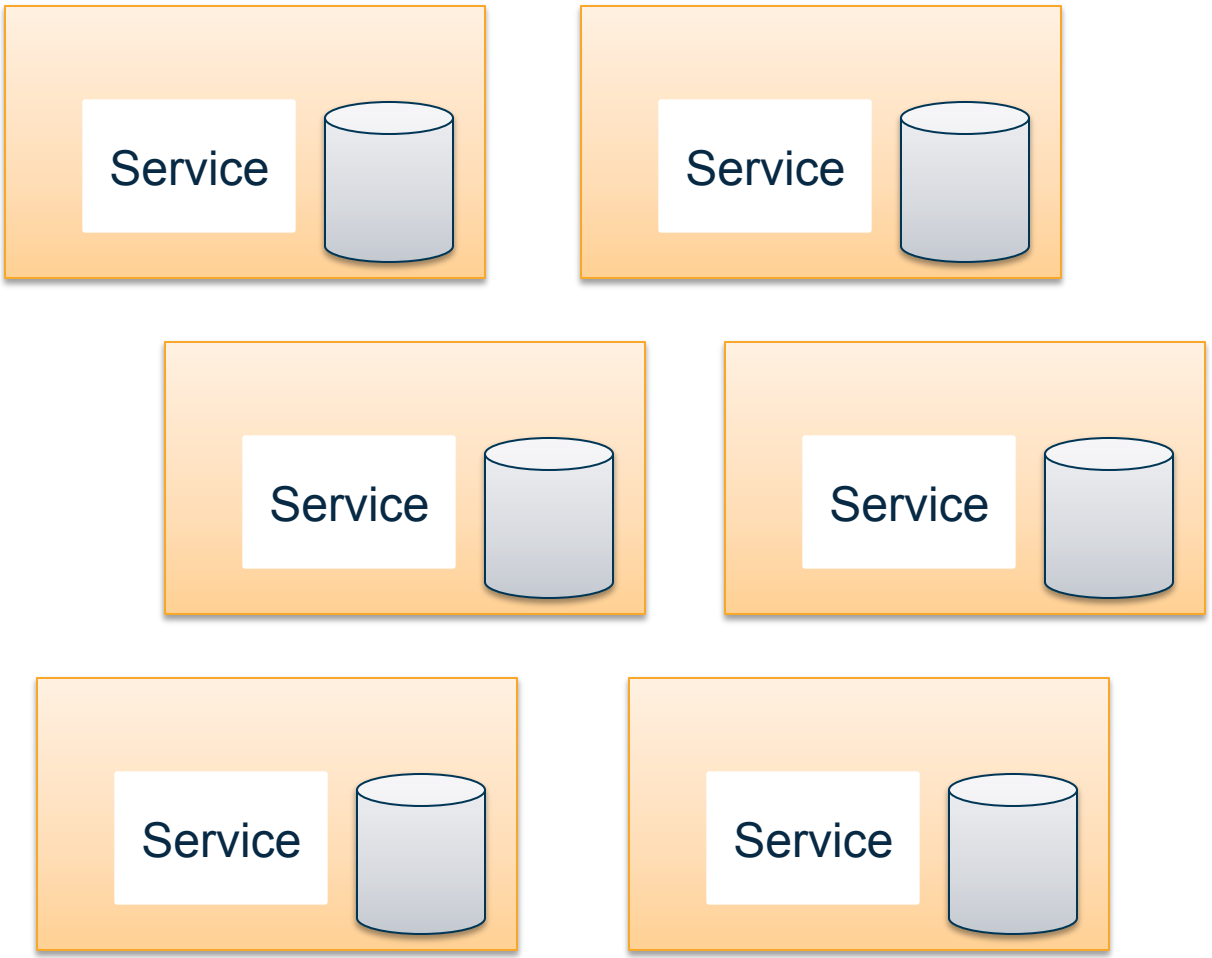
The Business Application Model

Allows for application programmers to leverage your proven and unique business logic

Examples:

- Sales Forecasting
- Inventory Control
- Resource Scheduling
- Policy Decision Logic
- Part Locator

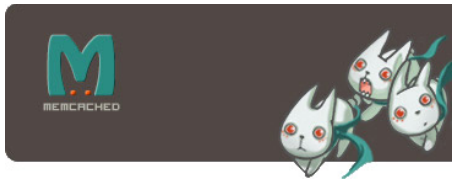
Thinking about Storage



NoSQL Databases

redis

Cloudant



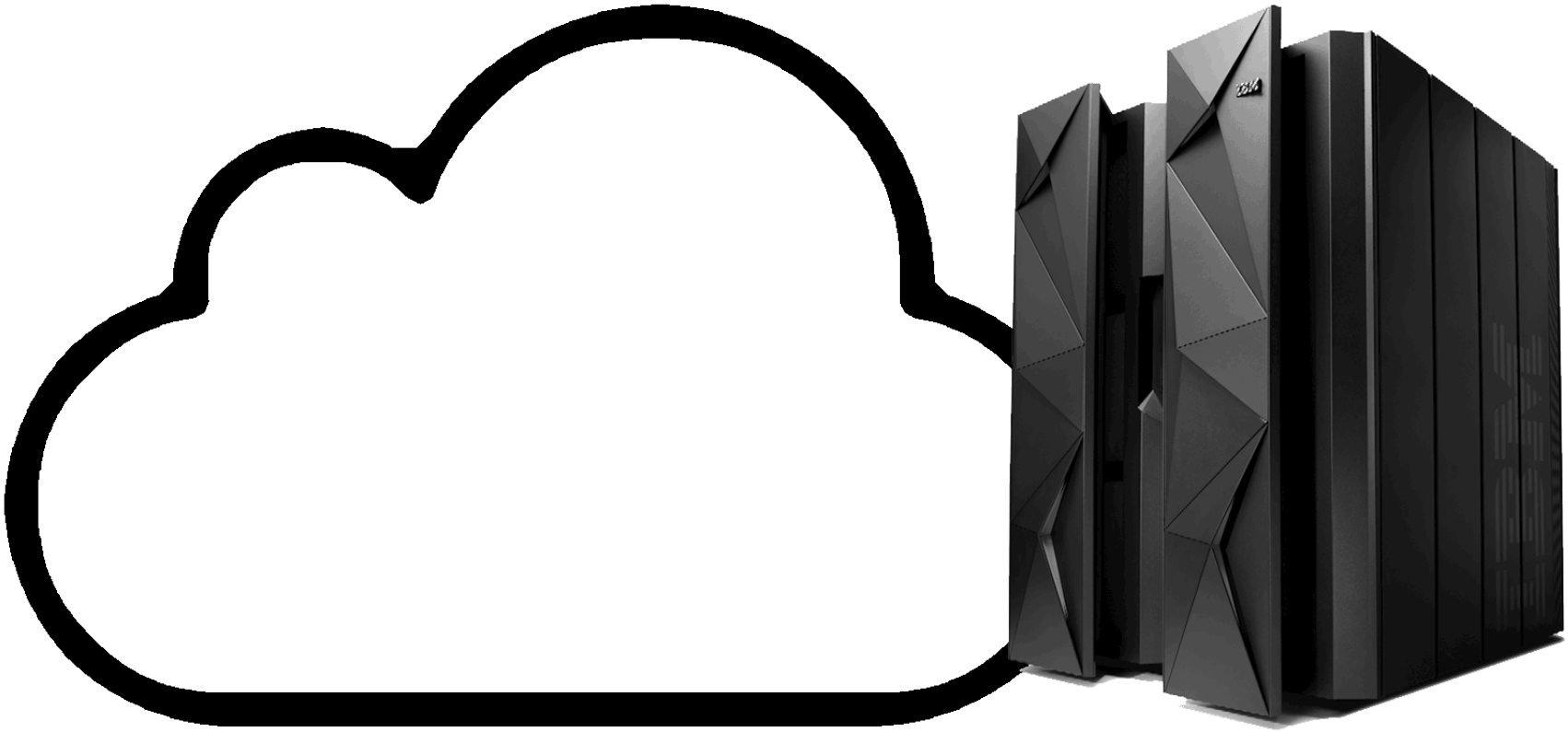
mongoDB

Remember those JSON objects?
Here's a good place to keep them.

InfoQ has released a video of Twitter's Kevin Weil speaking at Strange Loop earlier this year on how the company uses NoSQL. Weil is quick to point out that Twitter is heavily dependent on MySQL. However, Twitter does employ NoSQL solutions for many purposes for which MySQL isn't ideal. According to Weil, Twitter users generate 12 terrabytes of data a day - about four petabytes per year. And that amount is multiplying every year. Read on for our notes on Weil's talk.

Mainframe Considerations

Spoiler alert: It's just a big server



How do you define Multi-Tenancy?

Multi-user
Multi-consumer
Multi-instance
Multi-system
…

Public / Public / Hybrid Cloud Considerations

Another thing whose definitions have gotten hijacked lately

- **Private cloud** – Doing all of the cloud stuff just for my own use. Can be on or off-prem, and even provided by a third-party

- **Community cloud** – Built for a collective of users with similar demands and concerns. Ex: Healthcare providers using claim-specific service

- **Public cloud** – Owned by anyone, available to anyone

- **Hybrid cloud** – Two or more distinct cloud infrastructures which remain ubiquitous, but are leveraged to create applications and capabilities by using standardized or proprietary technology.

Hybrid Cloud

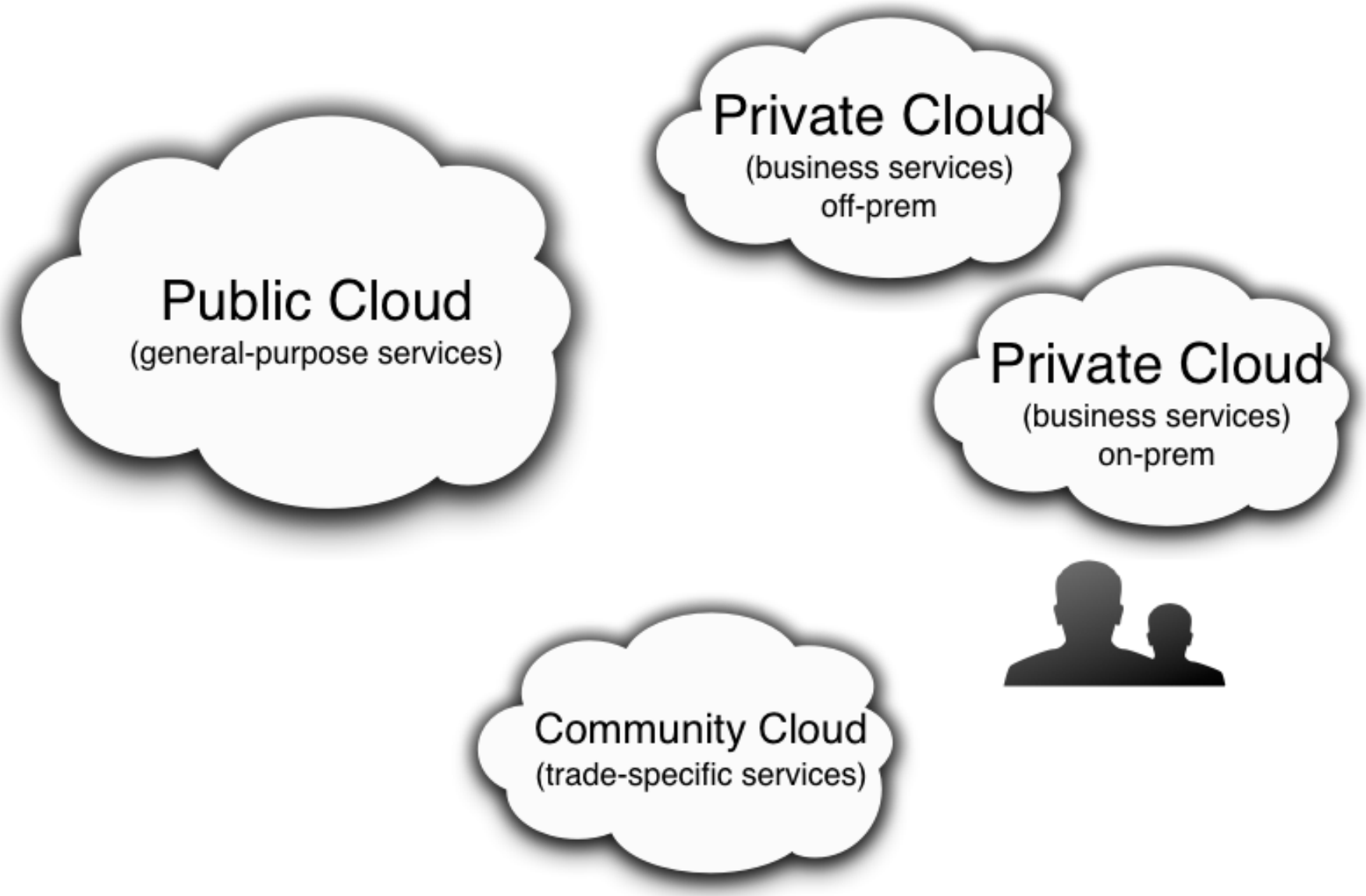
You keep using that word…

Hybrid cloud - *n.*

Two or more distinct cloud infrastructures which remain ubiquitous, but are leveraged to create applications and capabilities by using standardized or proprietary technology.

Hybrid Cloud

… with a chance of light precipitation in the evening



The API Economy

I'm not just doing this for my health

Local Connectivity

Phone numbers with prefixes that indicate geography

Numbers	Text Messages over Local Numbers		Picture Messages over Local Numbers	
$1.00 per month	$0.0075 to receive	$0.0075* to send	$0.01 to receive	$0.02* to send

Short Code Phone Numbers

5-6 digit phone numbers that can send and receive messages with mobile phones

Numbers	Short Code Text Messages		Short Code Picture Messages	
$3K per 3 months	$0.005 to receive	$0.01* to send	$0.01 to receive	$0.02* to send

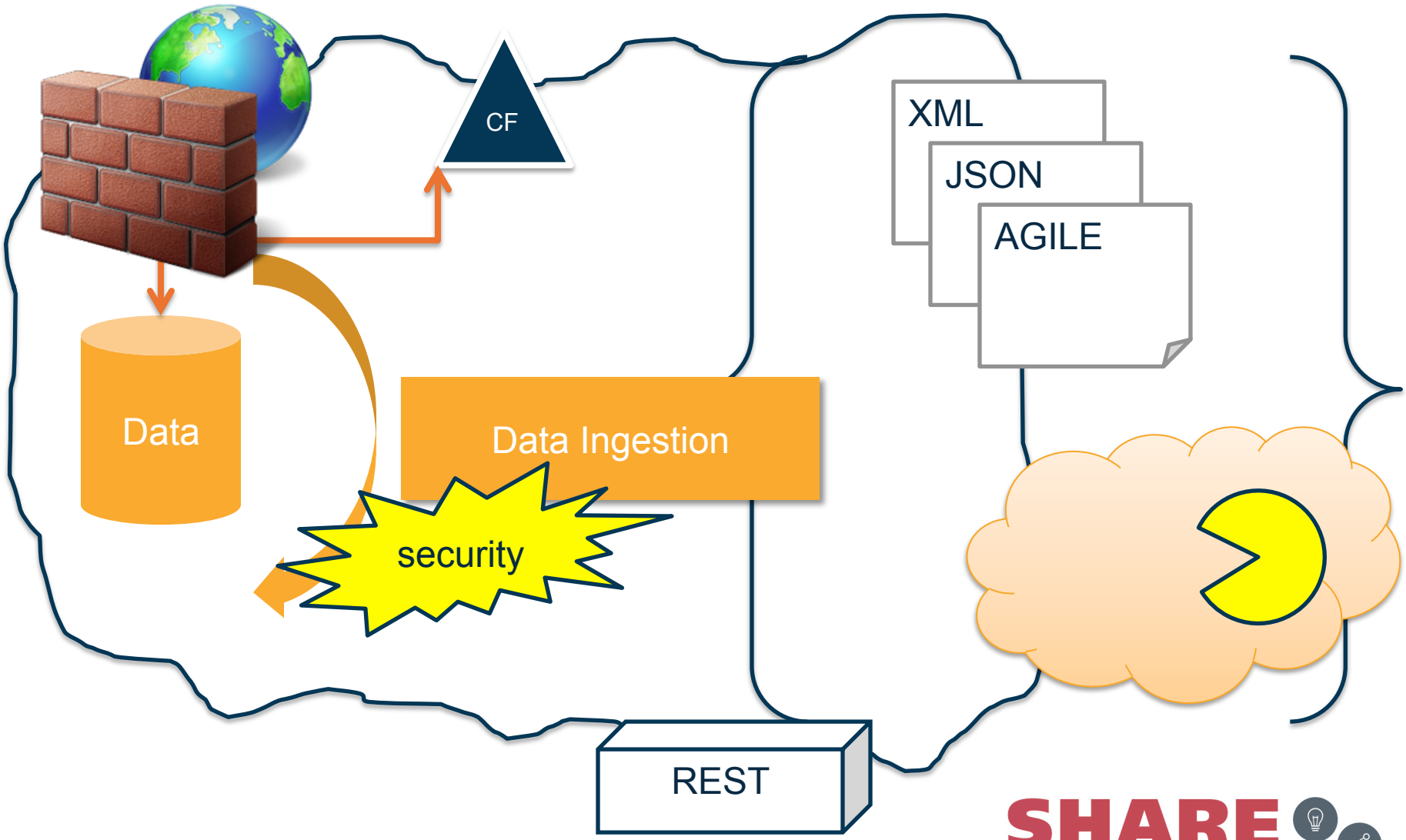
Toll-Free Phone Numbers

Phone numbers that are free for callers, making them ideal for your business

Numbers	Toll-Free Text Messages	
$2.00 per month	$0.0075 to receive	$0.0075* to send

Stop Chart Creep

Just because you made something in Visio…



Evaluating useability

Square pegs won't work for a round-hole business