

apigee



apigee

Introduction to Apache Usergrid: The BaaS built on Cassandra

Dave Johnson

Apigee

Agenda



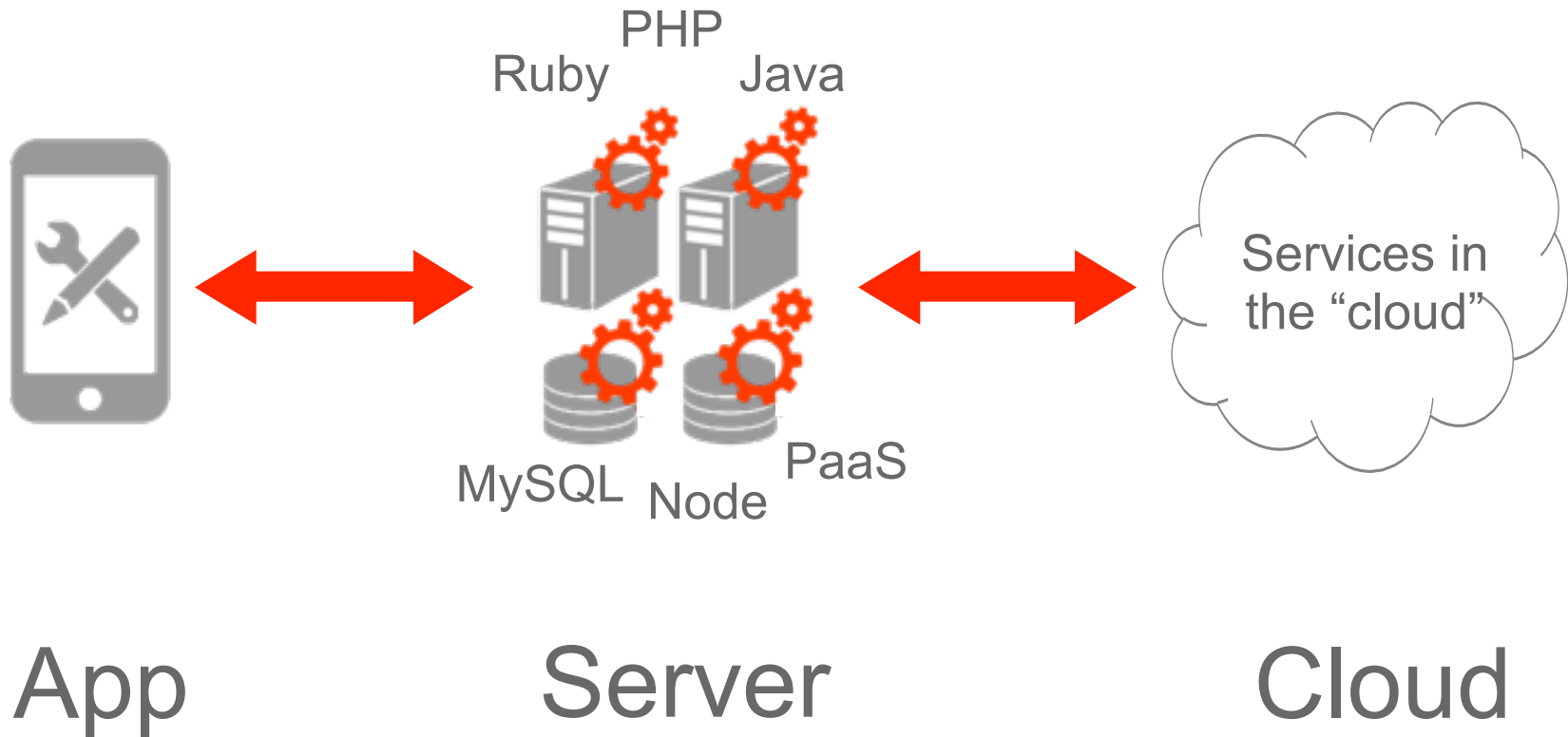
- » Overview: why Usergrid
- » Exploring the Usergrid API
- » Creating an app with Usergrid
 - » v1 - Add Logins & Signup
 - » v2 - Add Checkins
 - » v3 - Add Follower relationships
 - » v4 - Package & run via Apache Cordova
- » Running & Deploying Usergrid
- » Q & A

Usergrid Overview



Why Usergrid?

» Don't build a server



Why Usergrid?

» Focus on building your app



App



Cloud

What is Usergrid?



- » Mobile Backend as a Service (mBaaS)
- » Backend as a Service (BaaS)
- » Database as a Service (DBaaS)
- » REST API wrapped around Cassandra DB

What is Usergrid?

- » A Web Application
- » A Management Portal
- » Client SDKs
 - » iOS, Android, JavaScript, Node, Ruby, Java, Perl, .Net, Command-line, ...

What is Usergrid?

- » Core services you need to build apps:
 - » JSON object indexing & query
 - » Geo-location & geo-queries
 - » Connections & Collections
 - » Users, Groups, Social Graph
 - » Authentication, Roles & Permissions
 - » Asset & File Management

- » Soon: Push Notifications

What is Usergrid?

The screenshot displays the Usergrid Admin Portal interface. The browser address bar shows the URL `10.1.1.161:8080/portal/#/org-overview`. The page title is "usergrid_" and the user is logged in as "superuser@example.com".

The main navigation menu on the left includes:

- ORG ADMINISTRATION (selected)
- APP OVERVIEW
- USERS
- GROUPS
- ROLES
- DATA
- ACTIVITIES
- SHELL

The "Org Administration" page contains the following sections:

- Current Organization:** test-organization (ID: 51b3175e-2cbd-11e4-ba1b-53b429fe4bb5)
- Applications:** A table listing applications with an "Add New App" button.

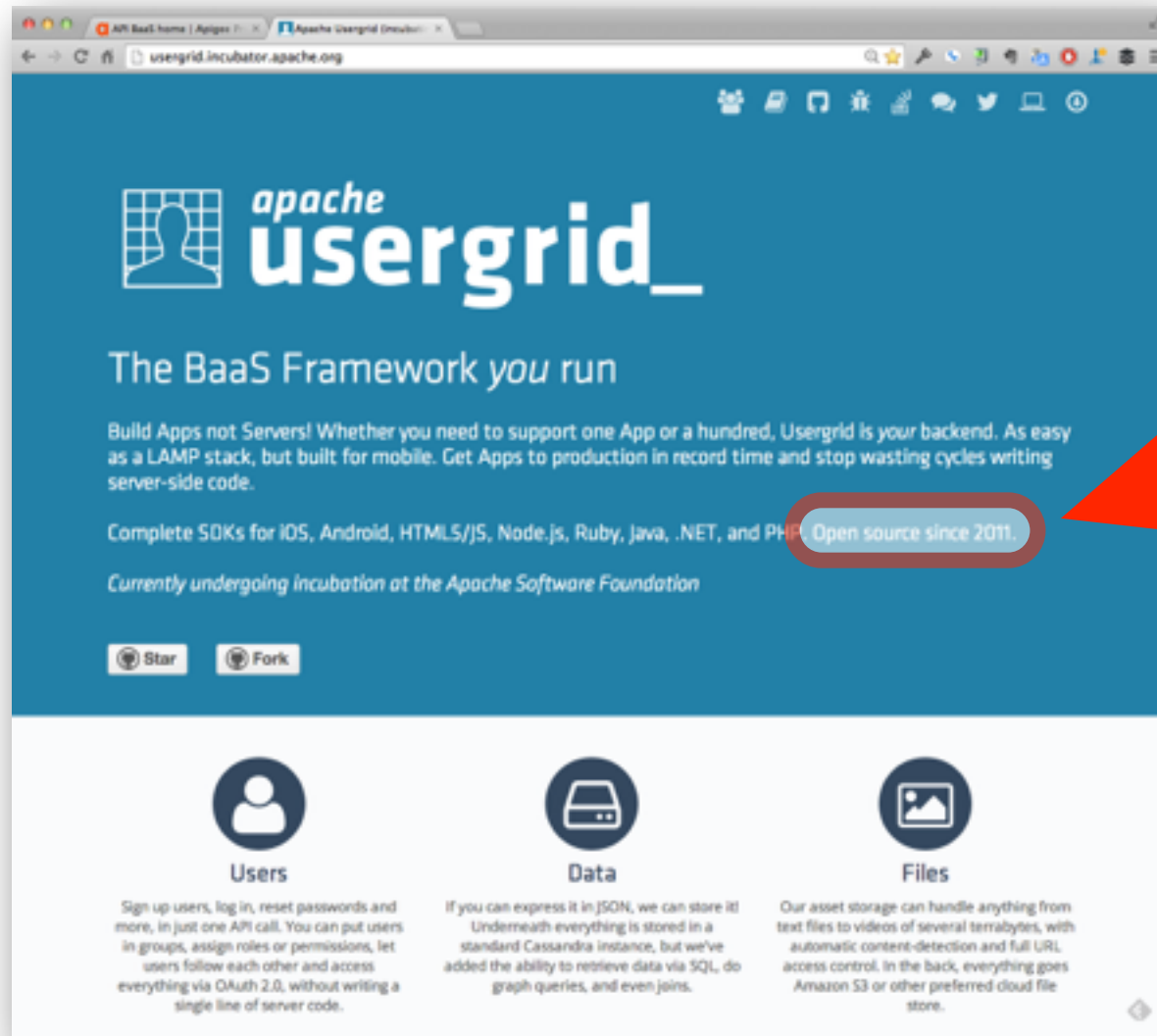
Application Name	ID
checkin1	32af6f20-3145-11e4-9050-174207aa18e2
restaurants	18d9b010-2dbf-11e4-a977-eb5245bda1cd
test-app	533b8c1a-2cbd-11e4-ab2a-ebca7b06267d
- Organization API Credentials:** Includes a "Regenerate Org Credentials" button and a table with Client ID and Client Secret.

Field	Value
Client ID	b3U6ubMXWly9Ee5e6G100K75L1Q
Client Secret	b3U6MHL3Bs7mx_ChZNGgCKTTZV6jY
- Organization Administrators:** Includes an "Add New Administrator" button and a table with administrator details.

Name	Email
Test User	testuser@example.com
- Activities:** A list of recent events, such as "Super User created a new application named checkin1" on Sun, 31 Aug 2014.

The footer of the page indicates "© Apache Usergrid 2014".

An Apache Incubator Project



The screenshot shows the Apache Usergrid website. The main heading is "The BaaS Framework you run". Below it, the text reads: "Build Apps not Servers! Whether you need to support one App or a hundred, Usergrid is your backend. As easy as a LAMP stack, but built for mobile. Get Apps to production in record time and stop wasting cycles writing server-side code." A red arrow points to a button that says "Open source since 2011.". Below this, it says "Complete SDKs for iOS, Android, HTML5/JS, Node.js, Ruby, Java, .NET, and PHP". At the bottom, there are three sections: "Users", "Data", and "Files", each with a brief description of its capabilities.

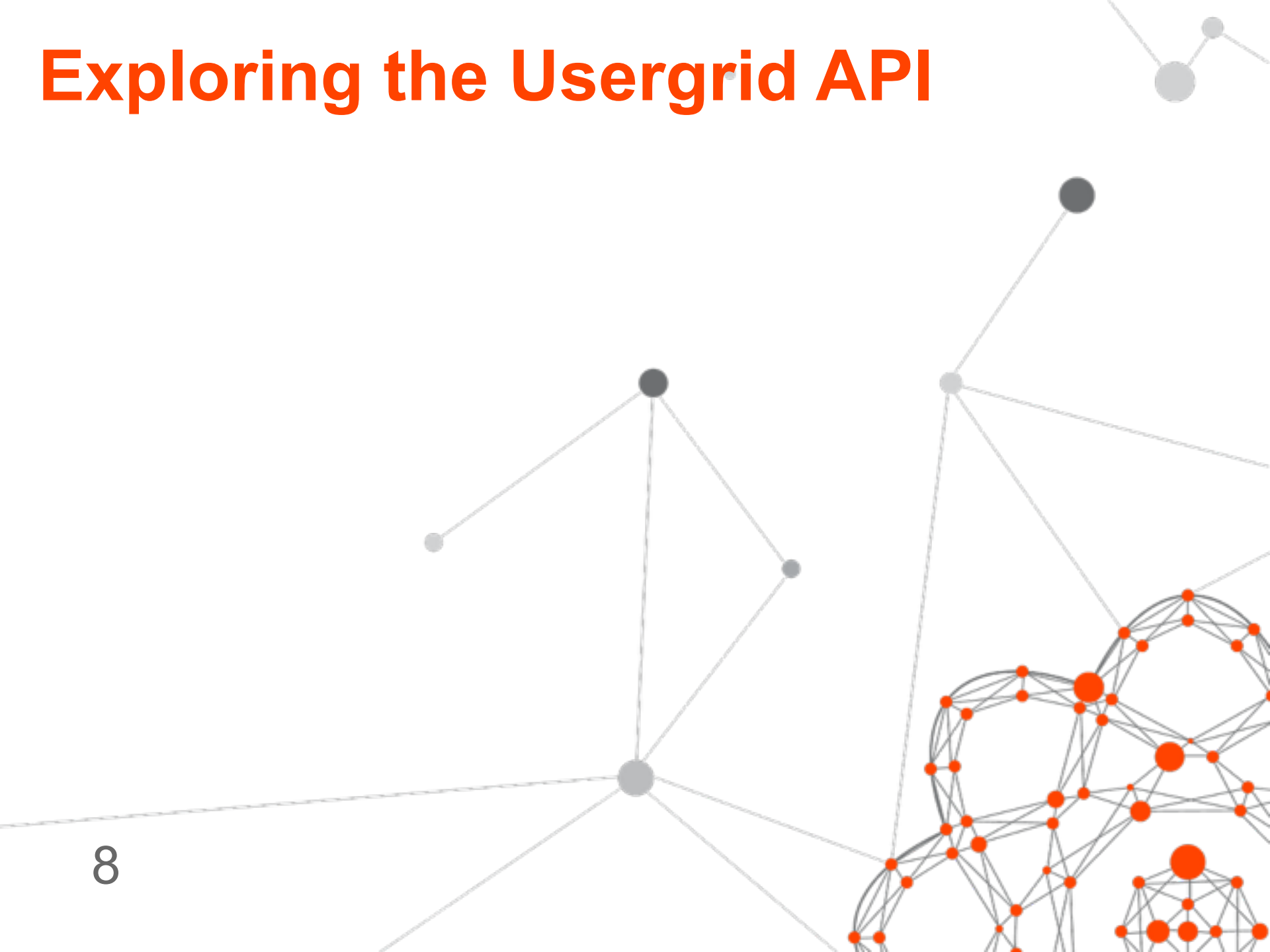
Star Fork

Users
Sign up users, log in, reset passwords and more, in just one API call. You can put users in groups, assign roles or permissions, let users follow each other and access everything via OAuth 2.0, without writing a single line of server code.

Data
If you can express it in JSON, we can store it! Underneath everything is stored in a standard Cassandra instance, but we've added the ability to retrieve data via SQL, do graph queries, and even joins.

Files
Our asset storage can handle anything from text files to videos of several terabytes, with automatic content-detection and full URL access control. In the back, everything goes Amazon S3 or other preferred cloud file store.

Exploring the Usergrid API



Usergrid REST API



- » JSON over HTTP
- » POST, GET, PUT and DELETE JSON Entities
- » Entities have UUID, Name, Type
- » Entities have name/value properties
- » All Entity properties are indexed
- » SQL-like syntax for Entity queries

Usergrid REST API

- » Each application has root URL like this
 - » `http:// host/{org}/{app}`
- » Entities exist in Collections
 - » `http:// host/{org}/{app}/{collection}`
- » And you can define Collections of custom Entities with any properties you want

Usergrid REST API

- » There are special built-in Entity Collections:
 - » `http:// host/{org}/{app}/users`
 - » `http:// host/{org}/{app}/groups`
 - » `http:// host/{org}/{app}/roles`
 - » `http:// host/{org}/{app}/activities`
 - » `http:// host/{org}/{app}/assets`

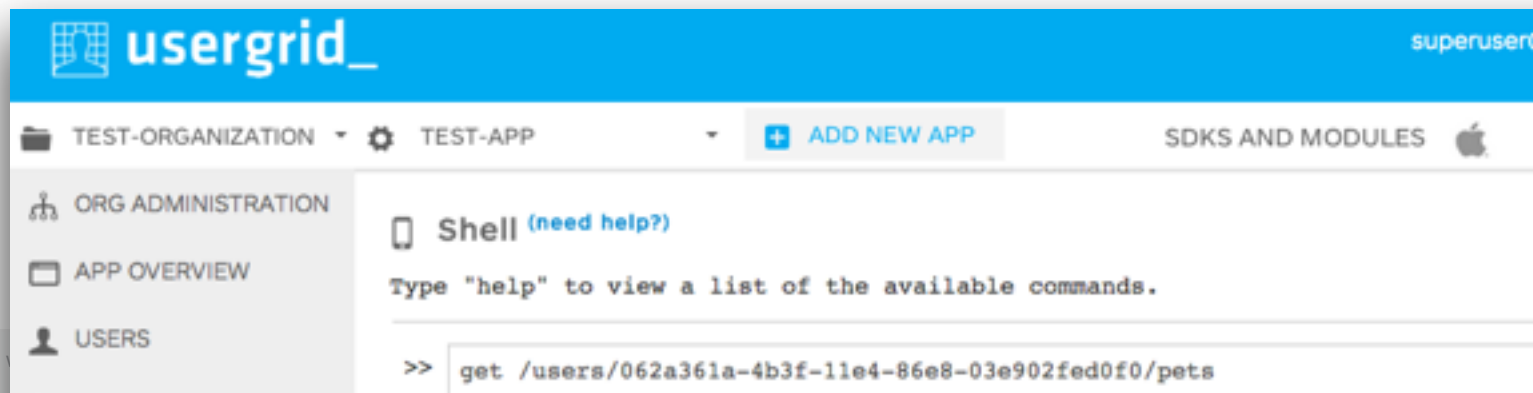
Connections



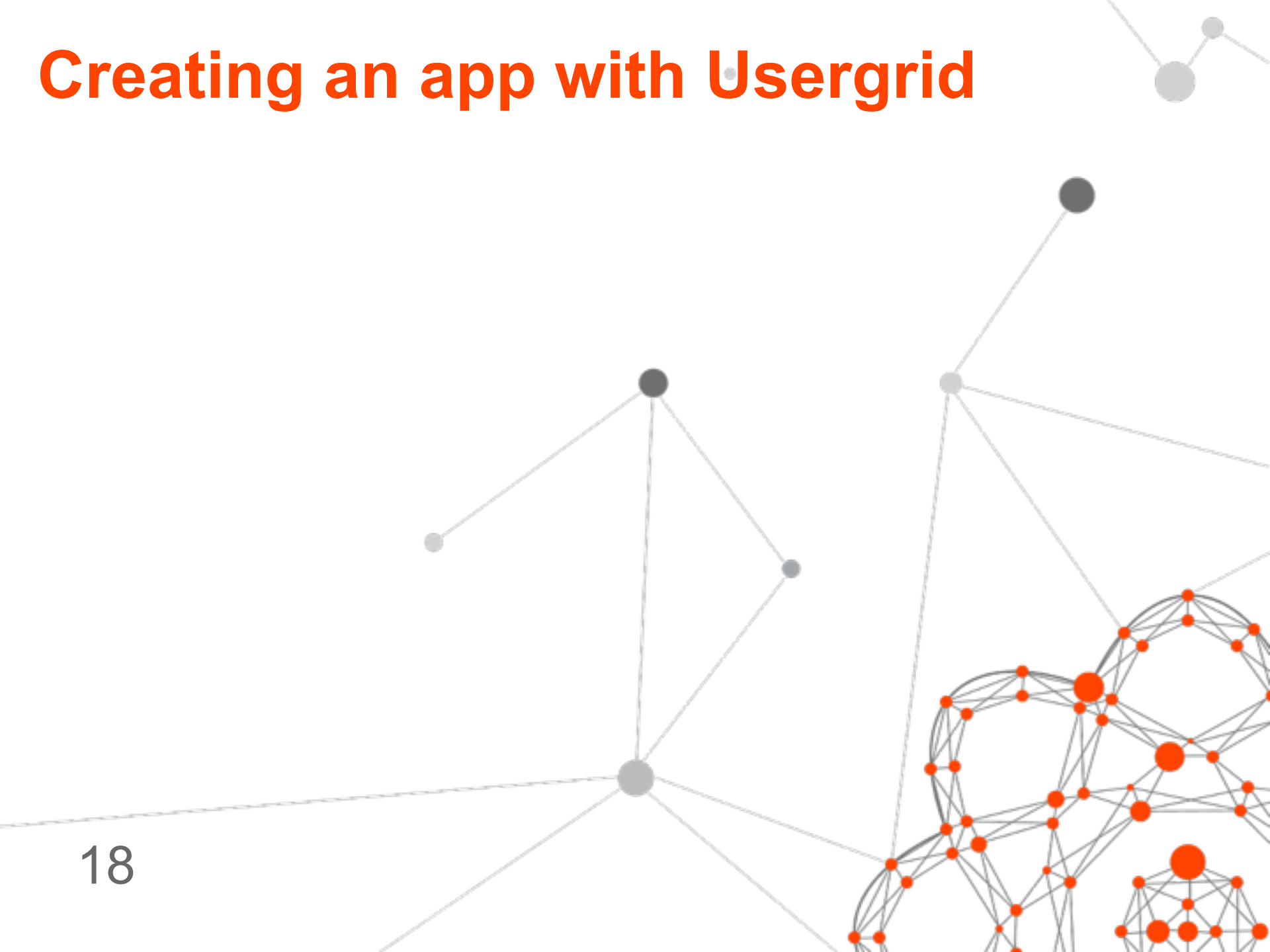
- » Entities can have Connections to other Entities
- » For example, to get all Entities that Dave is connected to via a connection of type “pets”
 - » `http:// host/${org}/${app}/users/dave/pets`

Collecting Cats

- » Demo the Portal
- » Create Application
- » Create User Dave with Default Role
- » Create Cats Collection
- » Setup “Dave has Cats” Connections



Creating an app with Usergrid



Let's create a checkin app

- » A simple checkin app to demonstrate Usergrid features:
- » Login & signup
- » Roles & Permissions
- » Following, Followers and Checkins
- » Usergrid JavaScript SDK
- » Built with jQueryMobile & Cordova

Define the data model

- » Usergrid provides the basic REST API
- » You must decide:
 - » Entity Types & Properties
 - » Entity Connections
 - » Users Roles & Permissions

Entities & Connections






- » Entity Collections
 - » Users
 - » Checkins

- » Entity Connections
 - » Users Follow Users
 - » Users Have Checkins

User Role: Guest

- » User who has not signed up yet
- » Permissions User with Guest Role:

	<i>GET</i>	<i>POST</i>	<i>PUT</i>	<i>DELETE</i>
<i>/users/*</i>				

- » Guest user POSTs to register
- » Usergrid takes care of account activation

User Role: Default

» Permissions for User with Default Role:

	<i>GET</i>	<i>POST</i>	<i>PUT</i>	<i>DELETE</i>
<i>/users/\${user}/**</i>	✓	✓	✓	✓
<i>/activities/*</i>	✓	⊘	⊘	⊘
<i>/users/*</i>	✓	⊘	⊘	⊘
<i>/users/\${user}</i>	✓	⊘	✓	⊘

App setup in the Portal



- » Quick demo
- » Take a look at Roles & Permissions
- » Ensure they are setup right

Adding Login & Signup

32

A network diagram consisting of several interconnected nodes and edges. The nodes are represented by circles of varying sizes. A cluster of nodes in the bottom right is highlighted in red, while the rest of the network is in grey. The red nodes are more densely connected to each other than the grey nodes are to each other.

Getting started



- » Pre-requisites:
 - » Usergrid JavaScript SDK
 - » jQuery Mobile
 - » Apache Cordova
 - » Your favorite dev environment



branch: master incubator-usergrid / sdks / html5-javascript / +



Add some missing headers, and make some header more consistent.

Dave Johnson authored on Sep 3

latest commit 17a9b8c309

..		
examples	Adding heads to files that need them.	2 months ago
extensions	Add some missing headers, and make some header more consistent.	a month ago
lib	Add some missing headers, and make some header more consistent.	a month ago
tests	Adding heads to files that need them.	2 months ago
Gruntfile.js	Removing repeated license headers from usergrid.js and adding it in b...	6 months ago
LICENSE	Create LICENSE	a year ago
README.md	Updated contribution section	4 months ago
Release.md	Adding release notes	8 months ago
changelog.md	Updated examples and fixed issues where encountered. Updated changelog	7 months ago
index.html	Add license headers to ALL HTML files in Usergrid	2 months ago
package.json	Revved the version to 0.11.0. Updated README and CHANGELOG	7 months ago
usergrid.js	Add some missing headers, and make some header more consistent.	a month ago

JavaScript SDK - login

```
var client = new Usergrid.Client({  
  appName: 'checkin1',  
  orgName: 'test-organization',  
  URI: 'http://10.1.1.161:8080'  
});  
  
client.login(username, password, function(err, res, entity) {  
  if (err) {  
    // inform user of error  
  } else {  
    // change to View Checkins page  
  }  
});
```



A Touch-Optimized Web Framework

jQuery Mobile is a HTML5-based user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices.

Download jQuery M

Custom download

Latest stable

Version 1.4.4

jQuery 1.8 - 1.11 / 2.1

Seriously cross-platform with HTML5

jQuery Mobile framework takes the "write less, do more" mantra to the next level: Instead of writing unique applications for each mobile device or OS, the jQuery mobile framework allows you to design a single highly-branded responsive web site or application that will work on all popular smartphone, tablet, and desktop platforms.

[Graded Browser Support](#)



Developer Links

- [Source Code \(GitHub\)](#)
- [jQuery Mobile Git \(WIP Build\)](#)
 - [JavaScript](#)
 - [CSS](#)
- [Report an issue](#)
- [Supported platforms](#)
- [Changelogs](#)
- [Upgrade Guides](#)
- [Roadmap](#)
- Discussion Forum
 - [jQuery Mobile](#)

JQuery Mobile page structure

```
<div data-role="page" id="my-page-id" data-theme="a">  
  
    <div data-role="header" data-theme="a">  
        // header goes here  
    </div>  
  
    <div data-role="content">  
        // page goes here  
    </div>  
  
    <div data-role="footer" data-theme="a">  
        // footer goes here  
    </div>  
  
</div>
```



Apache Cordova is a platform for building native mobile applications using **HTML, CSS** and **JavaScript**

[DOWNLOAD](#)

Cordova version 4.0.0



About Apache Cordova™

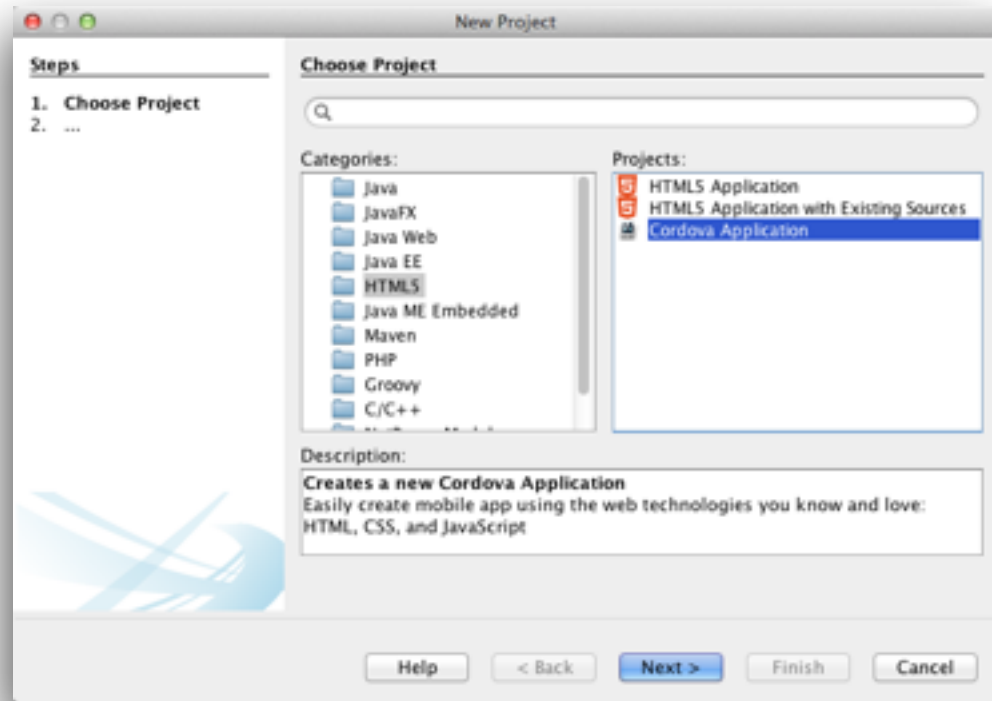


Apache Cordova

- » Hybrid approach to mobile apps:
 - » Write code in with HTML5 & JavaScript
 - » Package as a truly native app
 - » Access device features: camera, contacts, etc.
- » iOS and Android, but also Amazon Fire, Blackberry, Windows Phone and more
- » Tool support: Adobe, Eclipse, Netbeans, IntelliJ and even Microsoft Visual Studio

Get setup in an IDE

» Demo time



» Create Cordova project in IDE

» Add Login & Registration Page

What we'll add

» index.html

- » `$(document).on("mobileinit", ...`
- » `login-page`
- » `signup-page`
- » `checkin-list-page` (a placeholder)

» index.js

- » `login()`
- » `logout()`
- » `signup()`

Adding Checkins



Checkins as Activities

- » Usergrid supports notion of Activity
- » And supports a “Following” Connection
- » Users post Activities to their own Collection
 - » `POST /users/${user}/activities`
- » Users can see Activities those they follow
 - » `GET /activities`

JavaScript SDK - create checkin

```
var data = {
  type: "checkin",
  content: content,
  verb: "post",
  actor: { username: user.get("username") }
};
client.createUserActivity(user.get("username"), data,
function( err, response, activity ) {
  if (err) {
    // alert user of error
  } else {
    // send user back to View Checkins page
  }
});
```

What we'll add

- » `index.html`
 - » `checkin-list-page`
 - » `add-page`
 - » `view-checkin-page`

- » `index.js`
 - » `buildCheckinList()`
 - » `checkin()`
 - » `showCheckinPage()`

Add Follower relationships



36

Usergrid Social Graph



- » Usergrid provides Social Graph features including Following and Groups
- » Fred can follow Barney:
 - » `POST /users/fred/following/barney`
- » Fred can see his followers:
 - » `GET /users/fred/followers`

JavaScript SDK - follow user

```
var target = // user to be followed
var options = {
  method: 'POST',
  endpoint: '/users/' + me + '/following/users/' + target
};
```

```
client.request(options, function (err, data) {
  if (err) {
    alert("Unable to follow user " + target);
  } else {
    alert("Followed user " + target);
  }
});
```


What we'll add

» index.html

- » user-list-page
- » view-user-page

» index.js

- » buildUserList ()
- » showUserPage ()
- » followUser ()

Package & run your app

40

A network diagram with grey and red nodes and connecting lines. The diagram is sparse on the left and becomes increasingly dense on the right. A central grey node is connected to several other grey nodes. On the right side, there are several clusters of red nodes, some of which are larger than others, indicating a more complex or interconnected network structure.

Apache Cordova



- » Demo time
- » Use Cordova to launch the app in iOS

What we'll add

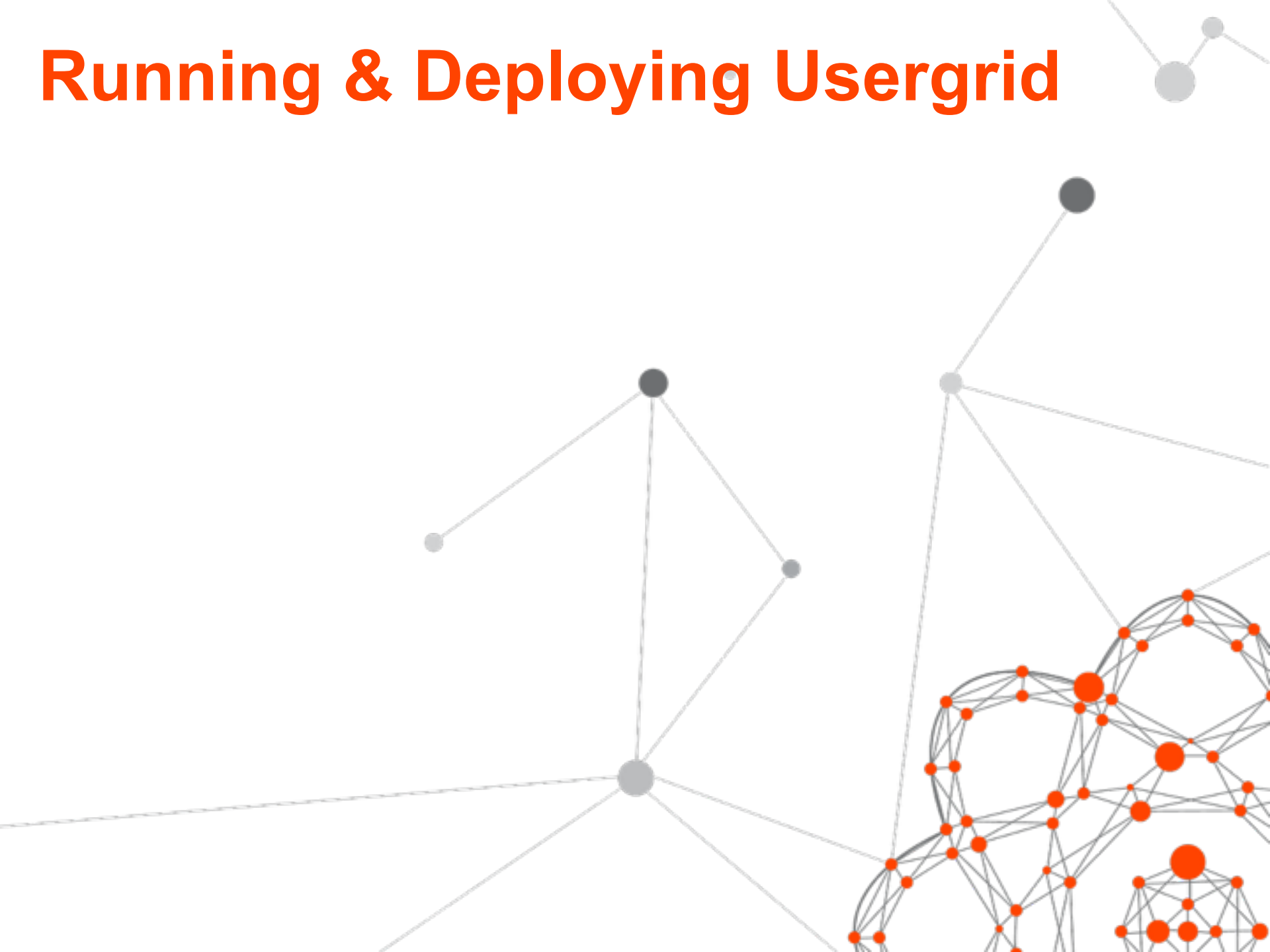
» index.js

» `deviceReady ()`

» `doWhenBothFrameworksLoaded ()`



Running & Deploying Usergrid



Developing with Usergrid

- » Three ways to run Usergrid locally:
 - » All-in-one Usergrid Launcher
 - » Install Tomcat, Cassandra and Usergrid
 - » Start Usergrid instance via Vagrant
- » Or find a BaaS provider that runs Usergrid:
 - » e.g. Apigee's API BaaS is free to use

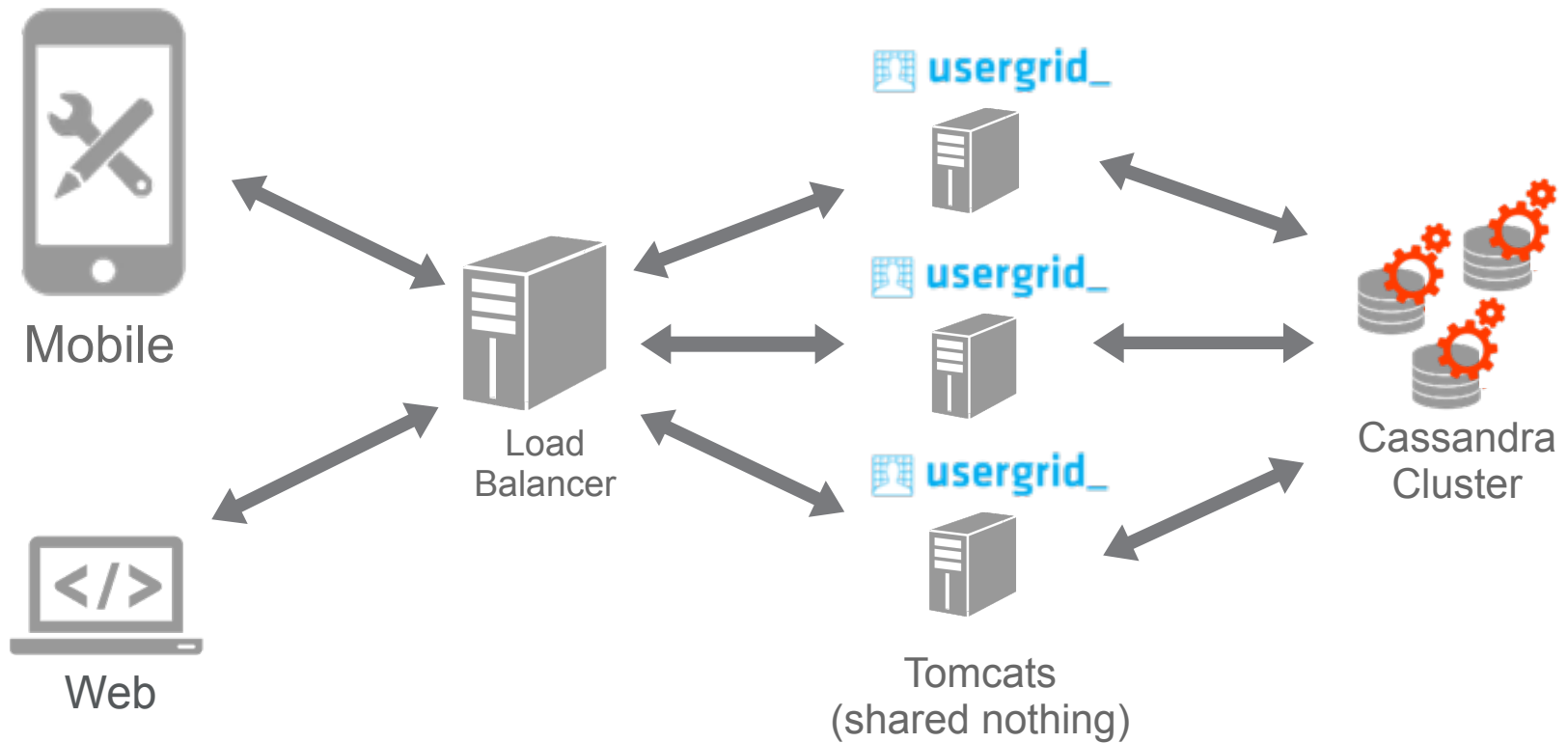


Usergrid deployment options

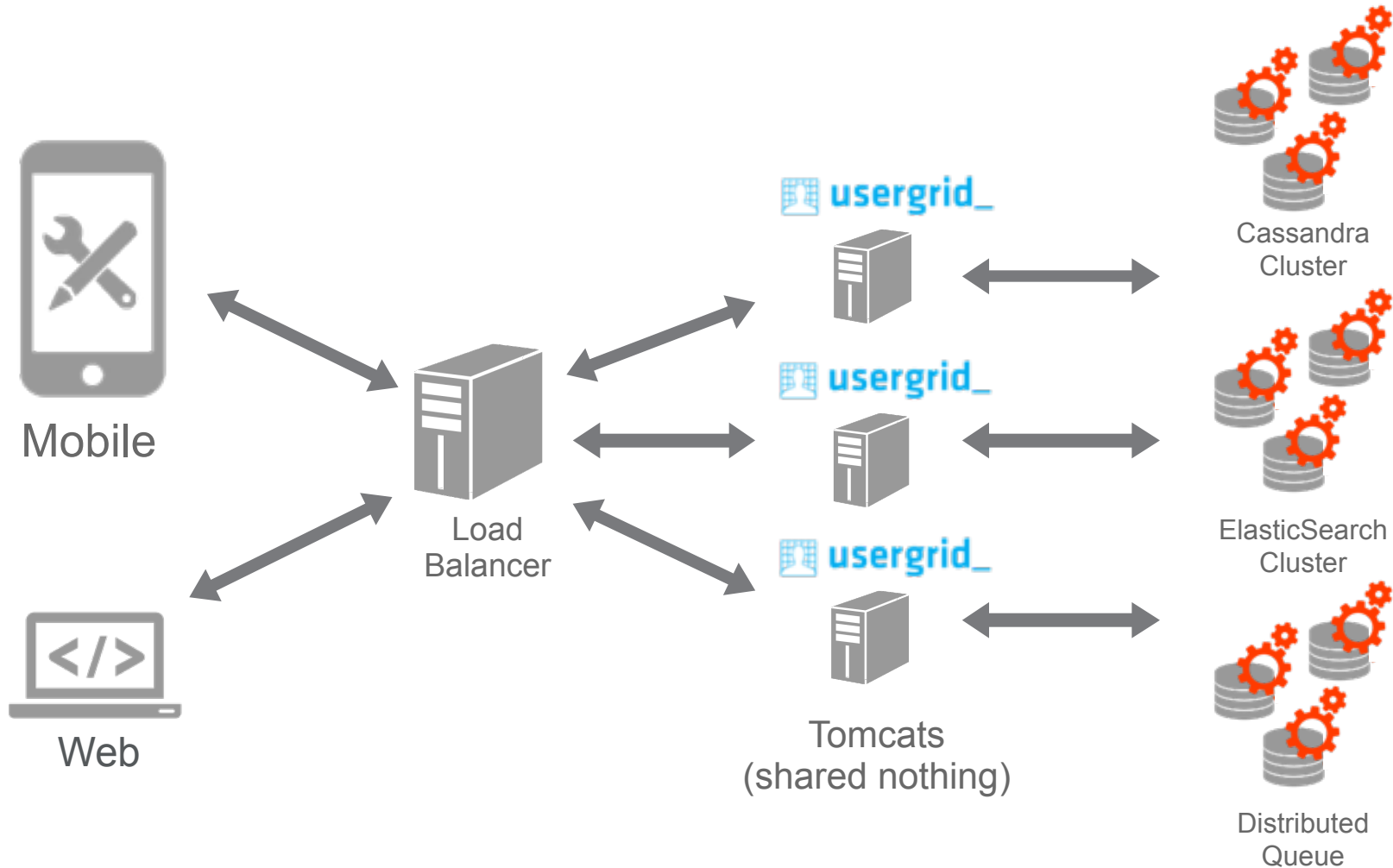
- » Do it yourself with Tomcat, Cassandra and Puppet, Chef, etc.
- » Do it yourself on AWS via **Usergrid AWSCluster**
 - » AWS Cloud Formation Template
 - » Plus supporting Bash and Groovy scripts
- » Let somebody else run it for you...
 - » Hint hint 😊



Deployment Architecture UG1



Deployment Architecture UG2



Wrapping up...



Questions?

- » Apache Usergrid
 - » <http://usergrid.incubator.apache.org>
- » Apigee API BaaS
 - » <http://goo.gl/zcjRxC>
- » Dave Johnson
 - » <https://github.com/snoopdave/>
 - » usergrid-mobile
 - » usergrid-vagrant
 - » <http://twitter.com/snoopdave>
 - » <http://rollerweblogger.org/roller>

Thank you



apigee