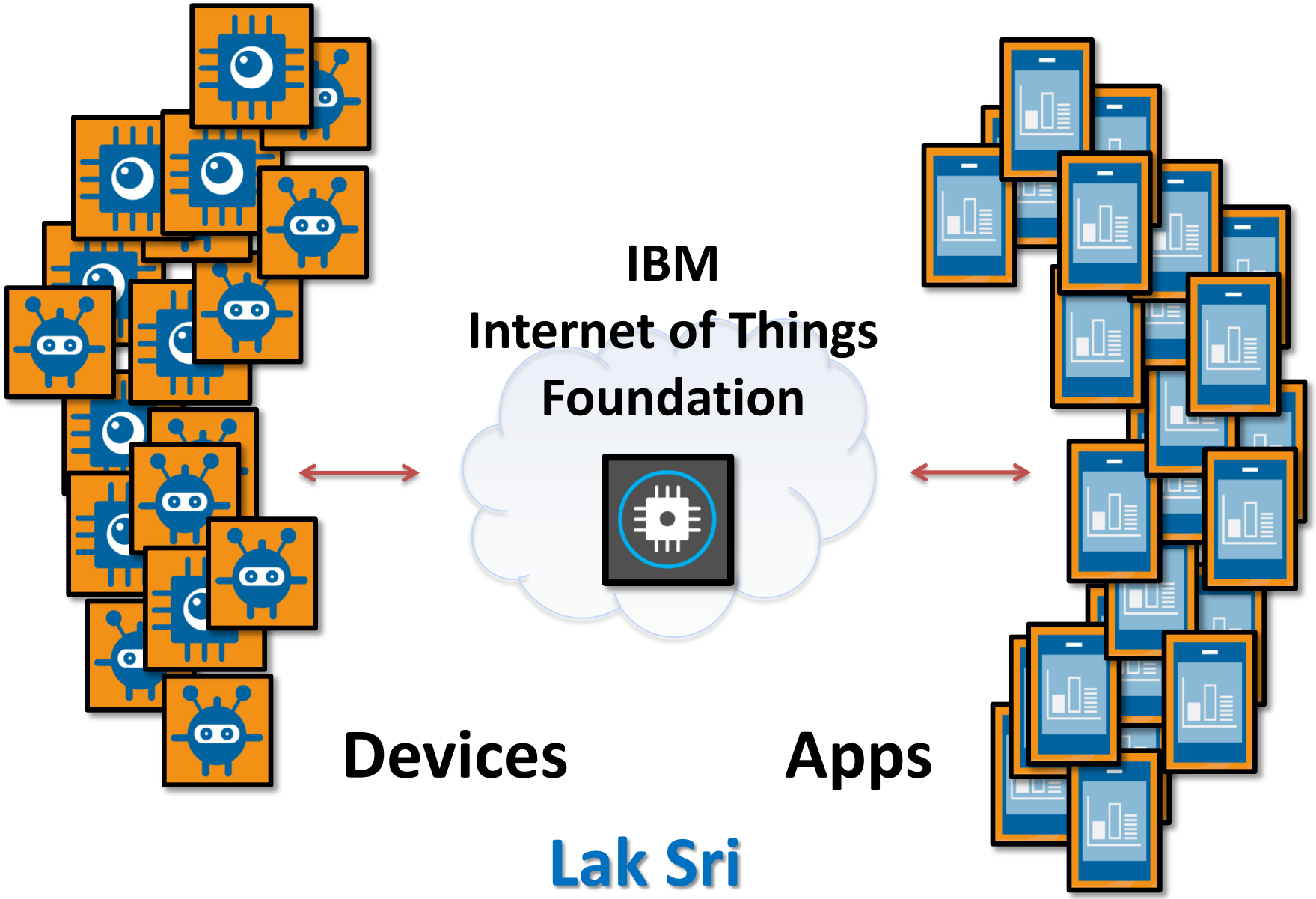


Lak Sri
laksri@us.ibm.com

IOT Applications

WHAT

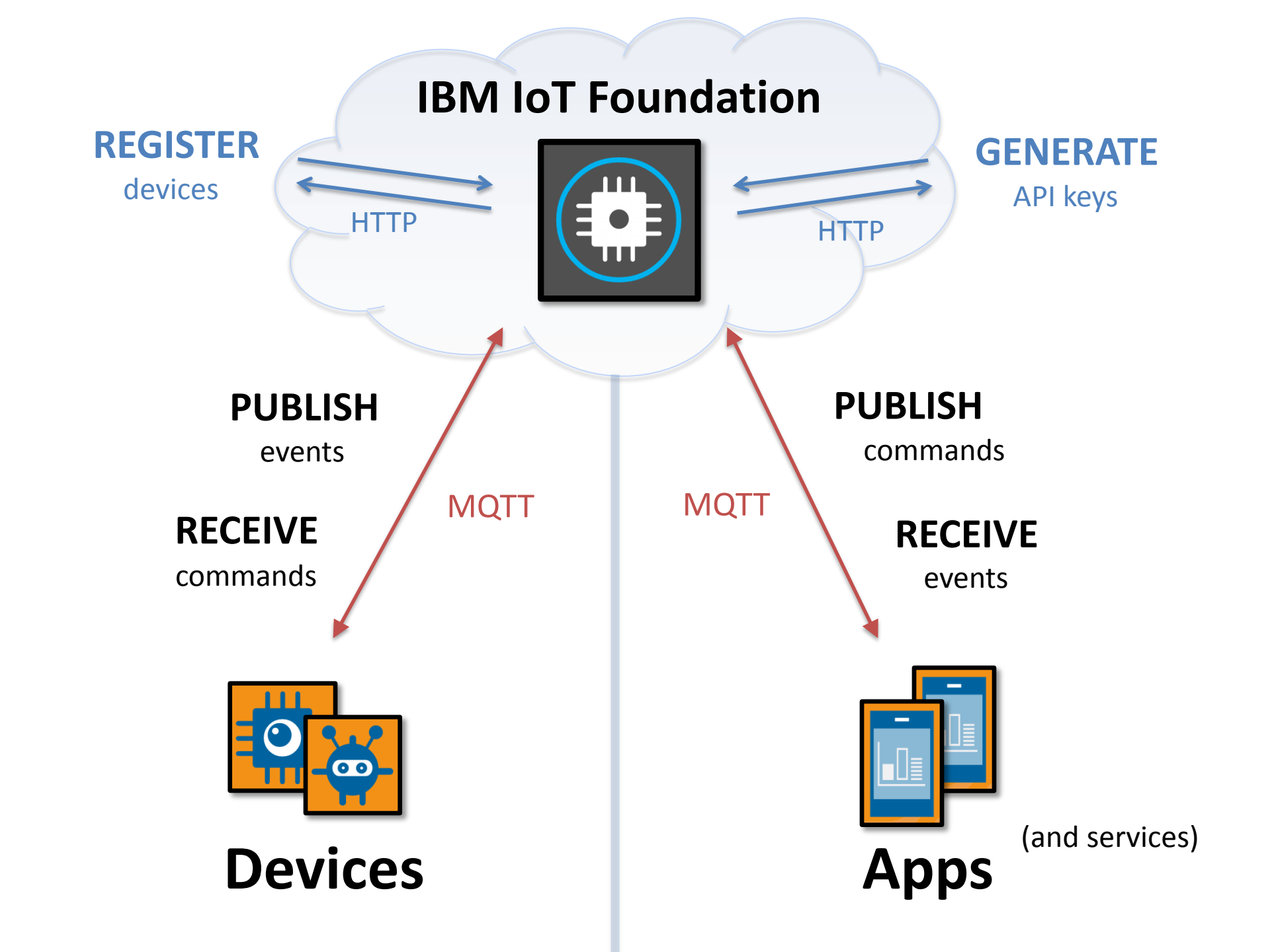
- Connect Real Devices or Simulate Virtual Devices Easily – IOT Demo

WHERE

- Reliable Platform – Scale to IoT scales – demo

HOW

- Great Visual Tool – Node-Red Demo


IBM IoT Foundation
REGISTER
devices
HTTP
GENERATE
API keys
HTTP
PUBLISH
events
RECEIVE
commands
MQTT
MQTT
PUBLISH
commands
RECEIVE
events
Devices
Apps
(and services)

Internet of Things

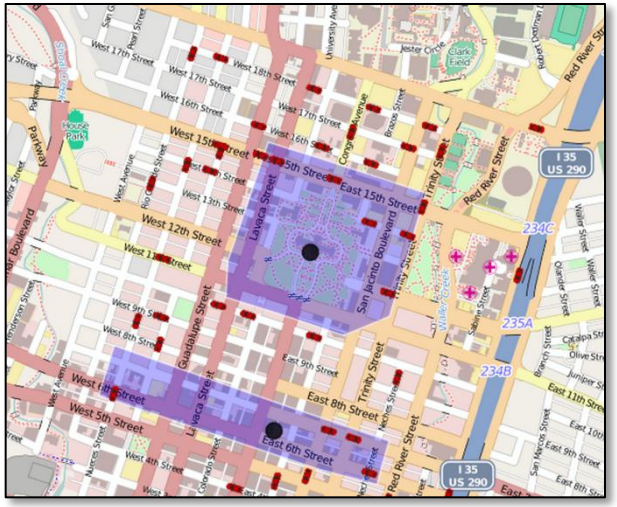
GOAL: simulate real-time traffic in Austin, send commands to vehicles, set up geofence alerts

Devices

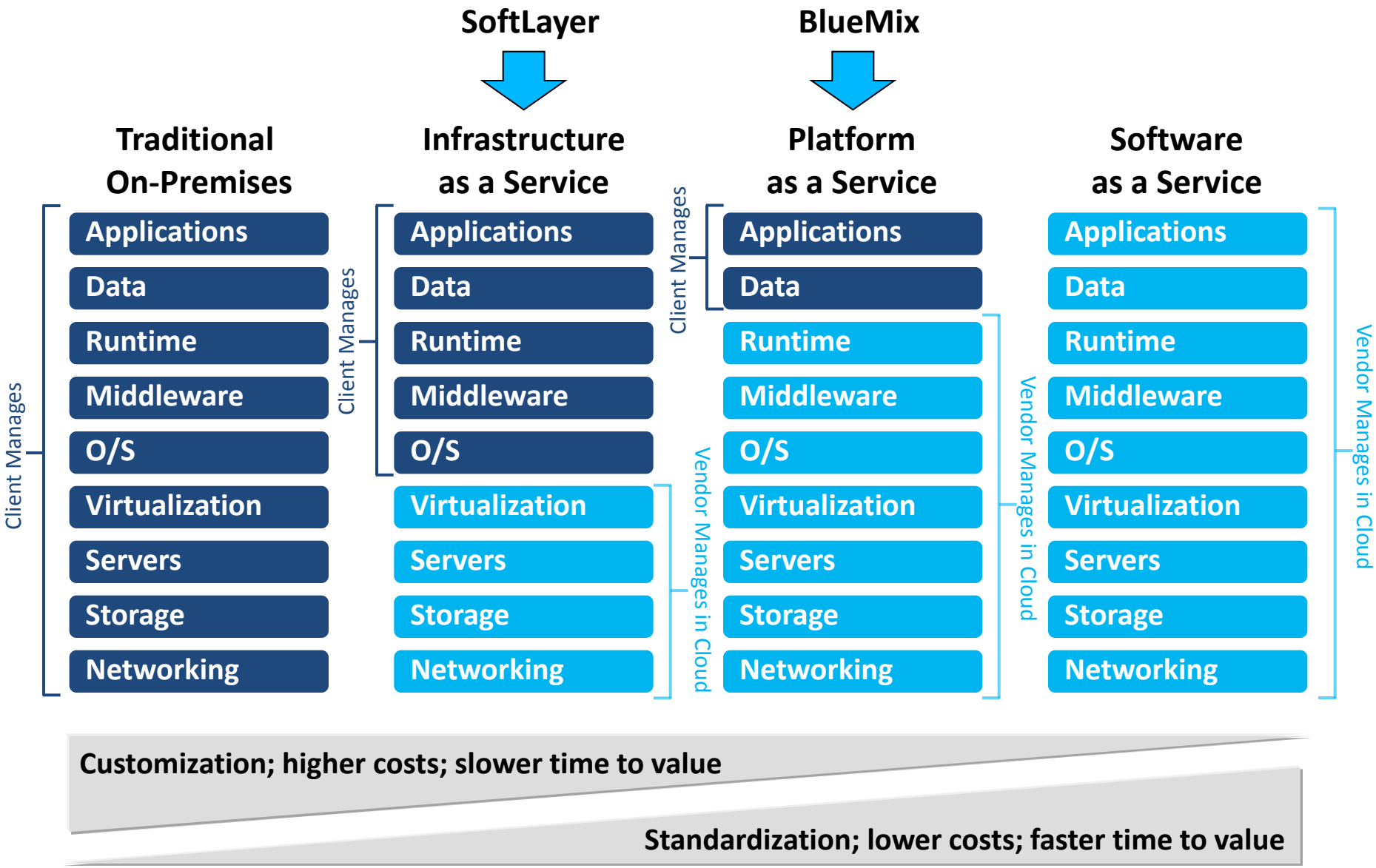
- Connected Car(s)

Applications

- TrafficView webapp
- Tester webapp
- Bluemix Geospatial Analytics

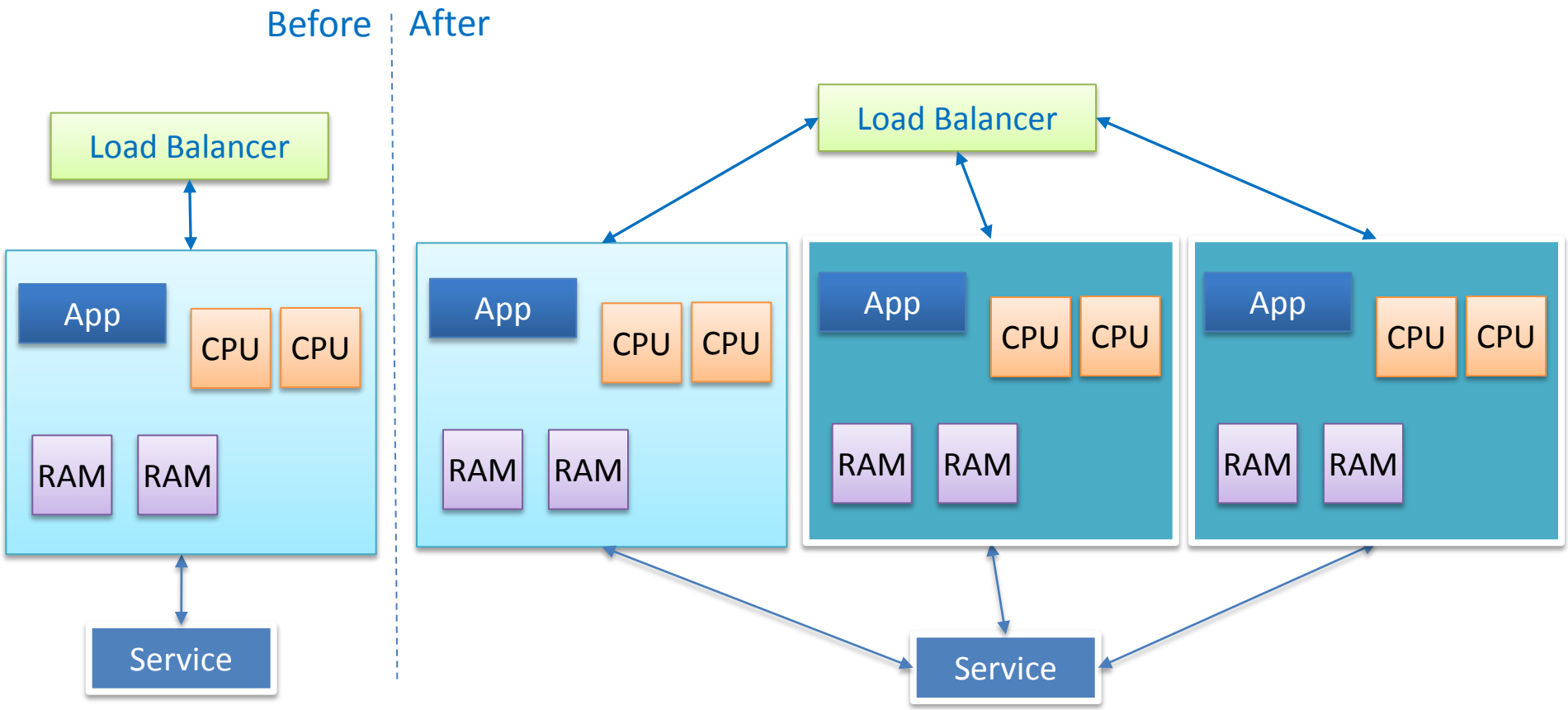


IBM Provided Cloud Service Models



Build on Scalable Platform

Increasing the number of nodes of the App Server through Load Balancing



Number of Instances is limited depending upon the cloud offering.

Increases Availability and Scalability
No changes to App required if written for PaaS / Cloud – but what are they

Building the IoT Application

0. (create a Bluemix account)
1. Set up an IoT Foundation account
2. Create placeholder app and services in Bluemix
3. Download and configure the application
4. Deploy to Bluemix with Cloud Foundry CLI
5. Play around with Node-RED and Geospatial Analytics

Step 1 – IoT Foundation

1. Sign up for account
   - http://internetofthings.ibmcloud.com
2. Create IoT organization
3. Register (two) devices
4. Generate API key

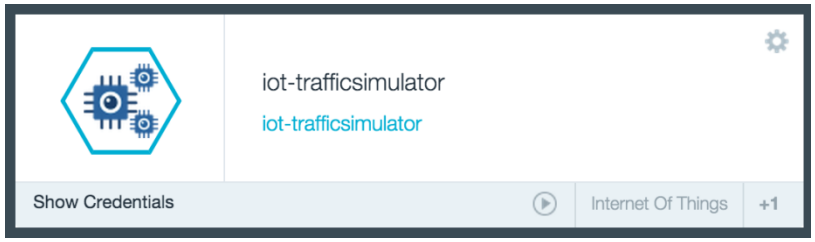
```
org=qdrlx
type=vehicle
id=ABC
auth-method=token
auth-token=I1GE+ZkC9GeBaTC)v0
```

```
org=qdrlx
type=vehicle
id=DEF
auth-method=token
auth-token=dI1E+Zkj9G2BaTa(4v
```

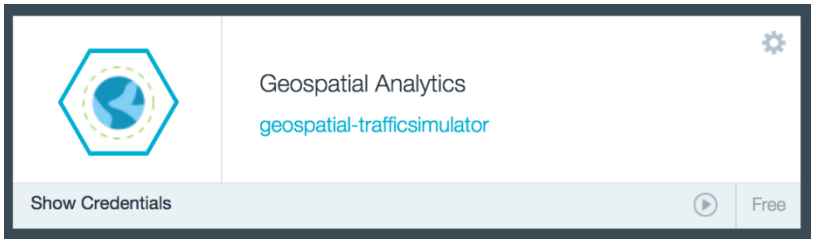
```
Key:          a:qdrlx:fizjs9ubu3
Auth Token:   c+fyOKEqPWf37gBA3)
```

Step 2 – Bluemix

1. Sign in to Bluemix account
2. Add application → Node.js runtime
   - name/host = <name>-trafficsimulator
3. (application) → Add a service → Internet of Things
   - enter your API key and auth token
4. (application) → Add a service → Geospatial Analytics



bryanboyd-iot-trafficsimulator



bryanboyd-geospatial-trafficsimulator

Step 3 – Download and Configure

1. Download: http://bit.ly/cc-trafficsimulator
2. Modify:
    1. manifest.yml
        - set host and name to match your Bluemix app
    2. public/config/settings.js
        - iot_deviceType = type from device registration
        - iot_deviceOrg = org from device registration
        - iot_deviceSet = id and auth-token for each device registered
        - iot_apiKey = API key
        - iot_apiToken = API token

```
org=qdrlx
type=vehicle
id=ABC
auth-method=token
auth-token=I1GE+ZkC9GeBaTC)v0
```

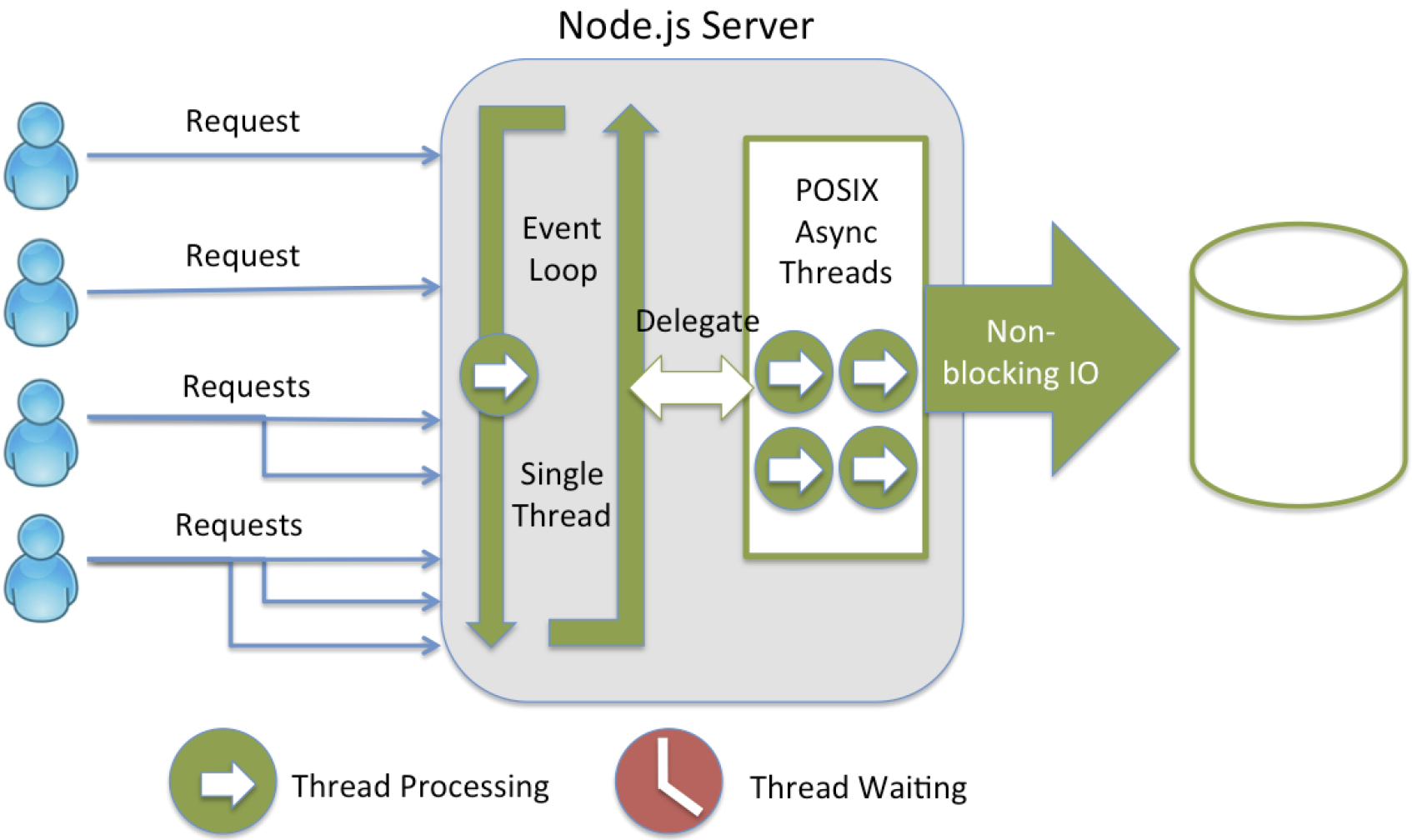
Step 4 – Deploy to Bluemix

1. Download and install CloudFoundry CLI tools:
   - OS X: http://bit.ly/cc-cf-osx
   - Windows: http://bit.ly/cc-cf-win
   - Linux: http://github.com/cloudfoundry/cli (find your distro) out)
2. Open prompt, navigate to trafficsimulator/
   - cf api https://api.ng.bluemix.net
   - cf login
     - Username> IBM ID username
     - Password> IBM ID password
     - Select an org... should be the same as your username
     - Select a space... probably **dev**
   - cf push <app name> (ex. bryanboyd-trafficsimulator)
3. When the application finishes staging, increase the simulation count with Bluemix app environment variable VEHICLE_COUNT

Relative I/O Latency – why any Non Blocking I/O of Node.js is benefitial

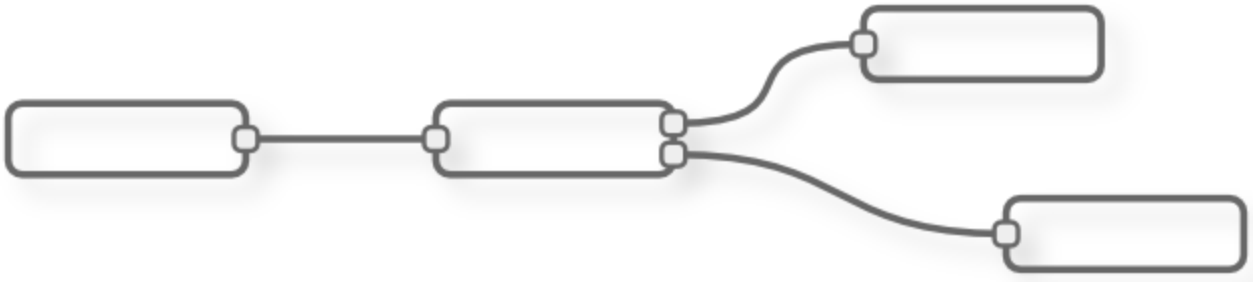
	CPU cycles	*relative*
L1 cache	3	next room ~5m
L2 cache	14	across the street ~20m
RAM	250	next block ~400m
disk	41 000 000	Earth circumference
network	240 000 000	distance to the Moon

Non Blocking - Node JS based IoT App for handling large number of connections

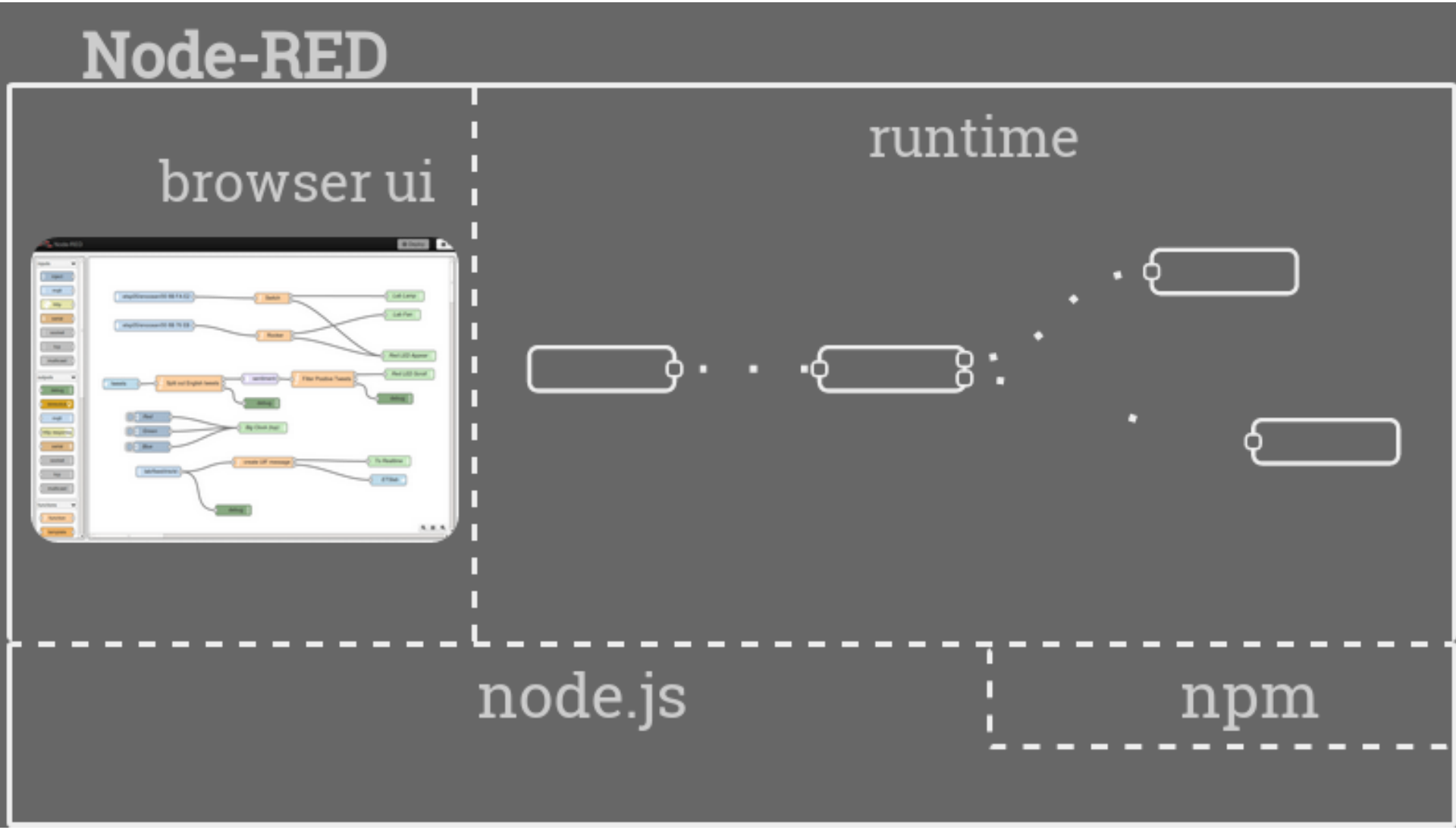


Node-RED

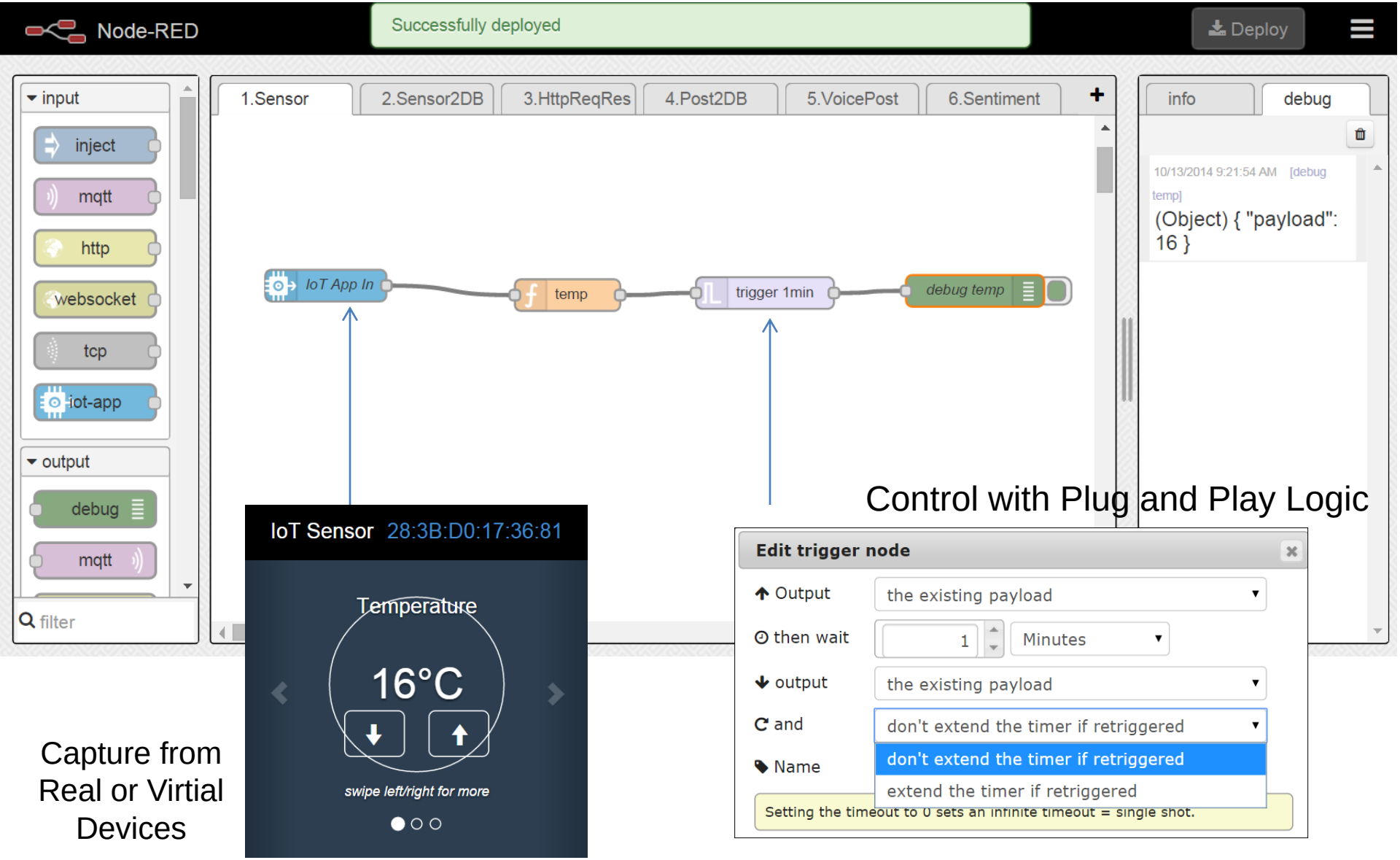
A visual tool for wiring the
Internet of Things
built on Node JS



Node-RED Design



Develop/Deploy/ Debug Node-RED Apps quickly



Control with Plug and Play Logic

Capture from Real or Virtial Devices

Use Database to save IoT data

Cloudant Lives at the Intersection of Big Data, Cloud, Mobile and Devices

- RESTful API – actually, there isn't any other interface than HTTP.
- JSON and JavaScript – it stores and serves JSON documents, as well as uses JavaScript to manipulate them during validation or querying.
- Multi-master asynchronous replication – documents can be bidirectionally replicated to many instances and every instance can simultaneously modify all of them.

Network Availability and Power consumption

IoT connectivity should be a forethought before deployment, not an after thought. Having reliable IoT network to connect devices and servers is critical for a large scale IoT app.

Reality is many times – we will have to use SMS for devices reporting movement and other information

Node-red Demo

Putting it all together

QA