



Loopback API Framework  
nodejs @ hyper-scale

Shubhra Kar | Products & Education

[twitter:@shubhrakar](https://twitter.com/shubhrakar)

# About me

- ◆ J2EE and SOA architect



- ◆ Performance architect

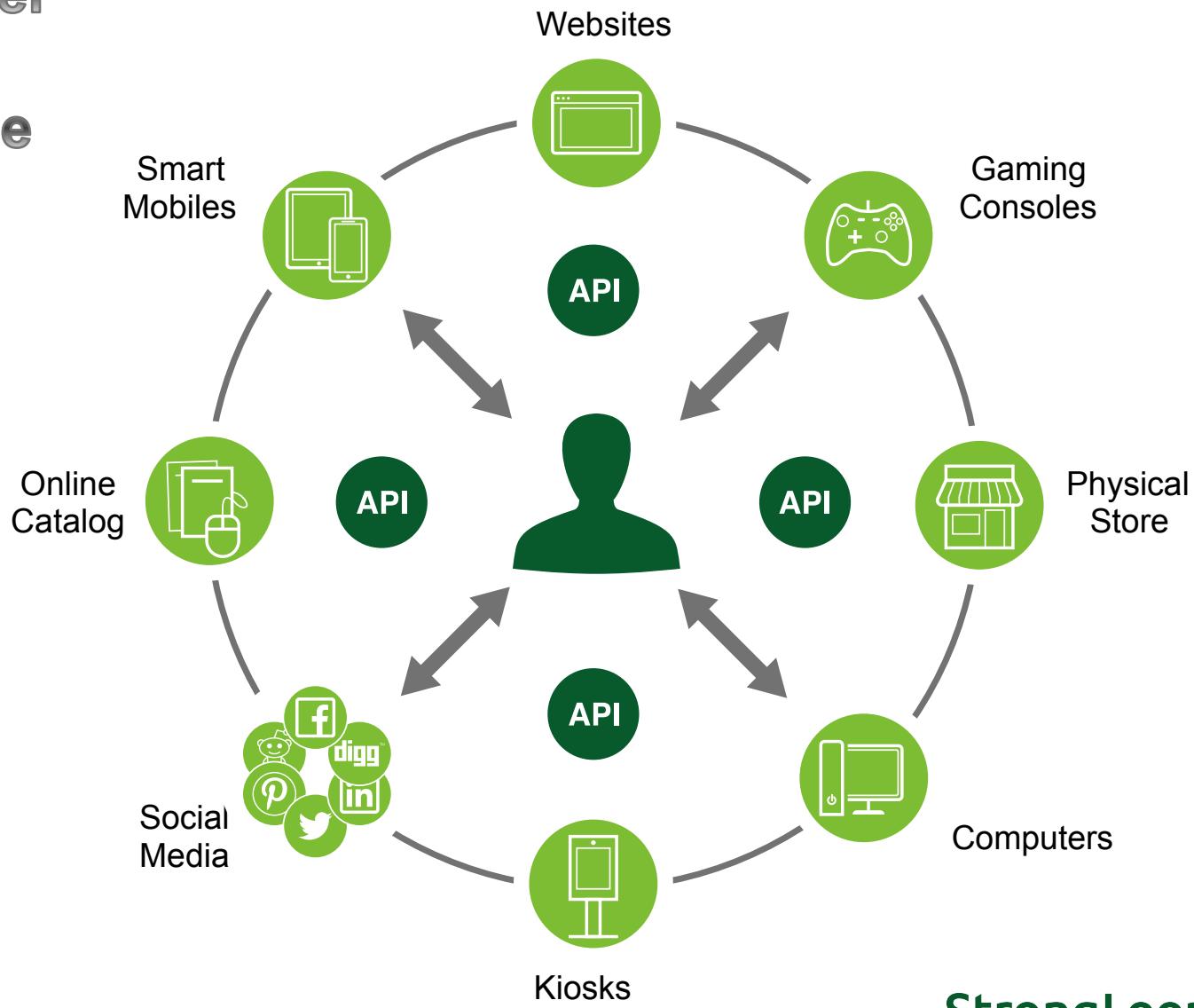


- ◆ Node, mBaaS & APIs

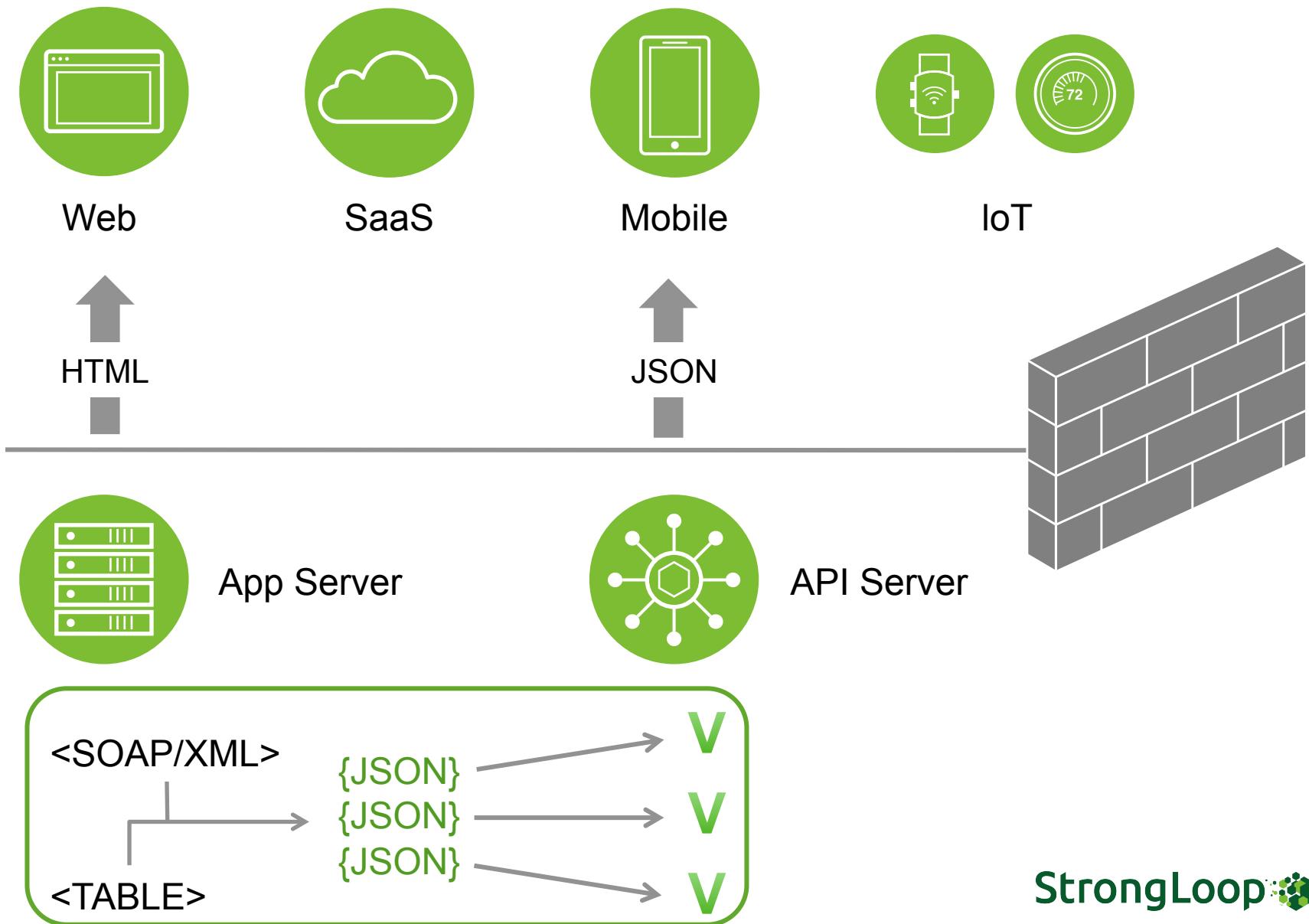


# Node.js API Buzzwords

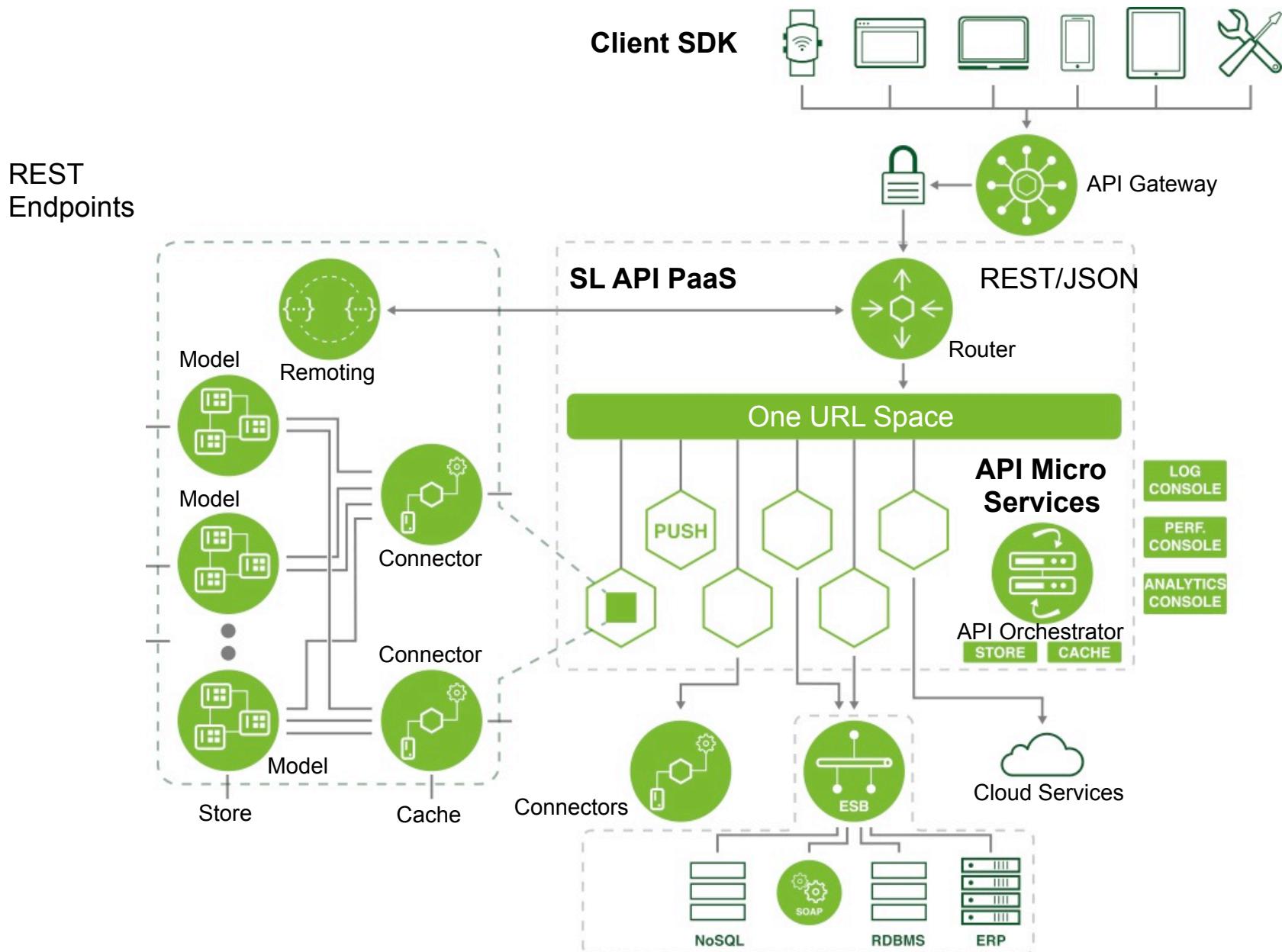
- Omni channel
- Isomorphic
- APIs At Scale



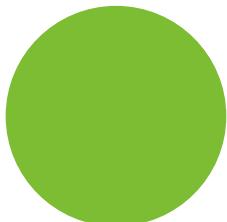
# API “Decomposition” is the game changer



# Node powered Micro-services has arrived



# Frameworks evolve as well !



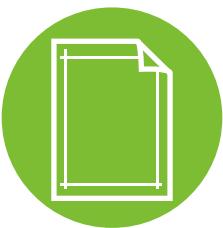
## KISS

- Callback
- Reactor
- Observer



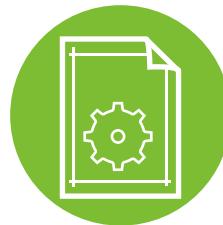
## MVC

- Express
- HAPI
- Restify
- Total
- Partial



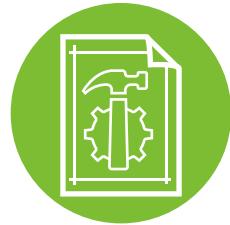
## MBaaS

- Loopback
- Parse
- Meteor
- Feedhenry



## ORM/ MEAN

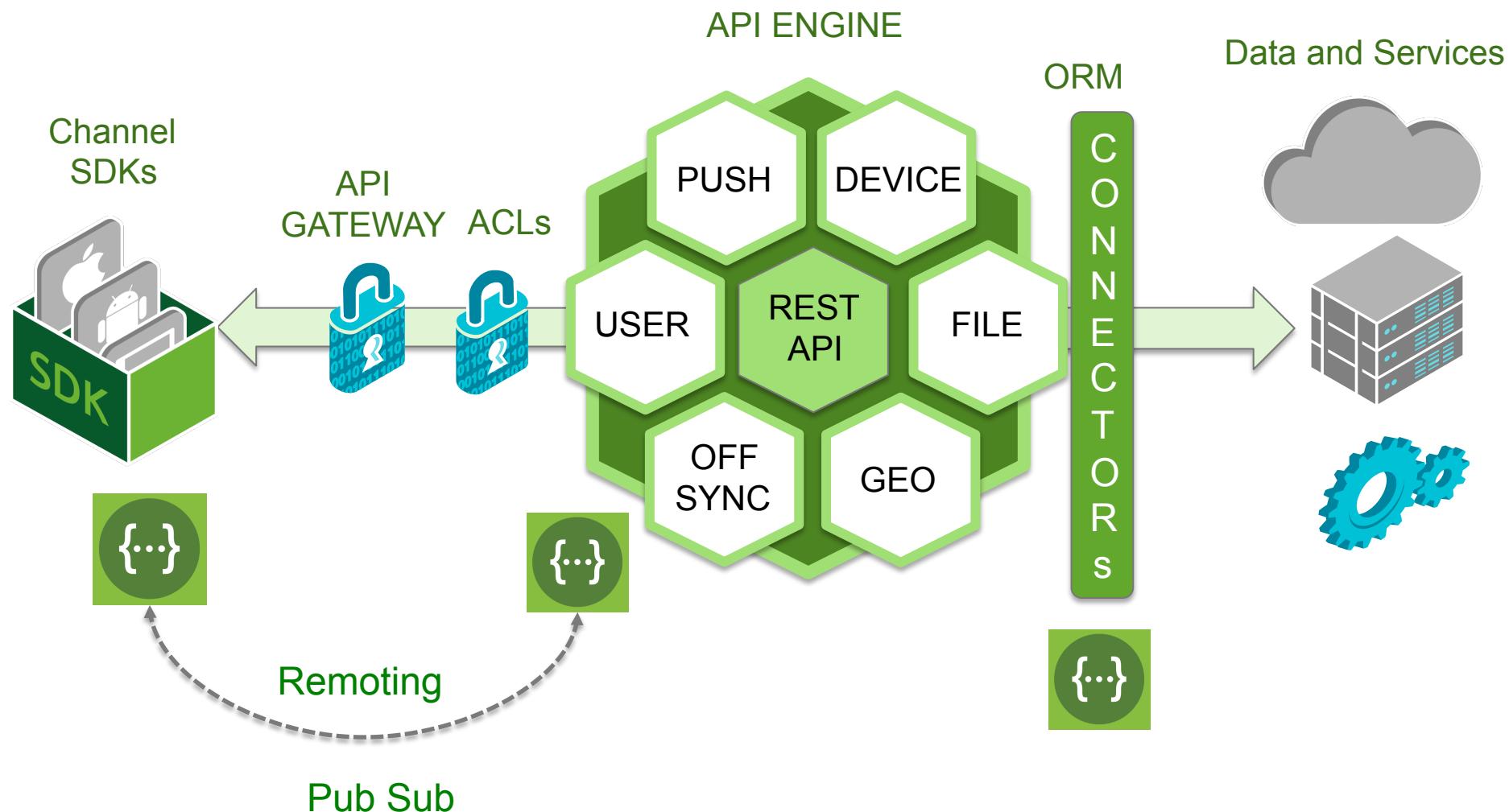
- Loopback
- Sails
- Meteor
- Geddy



## Micro/ IoT

- Loopback
- NodeRED
- Proprietary

# Loopback: Open Source nirvana (mBaaS, ORM, Micro)



# Right interface for the right user

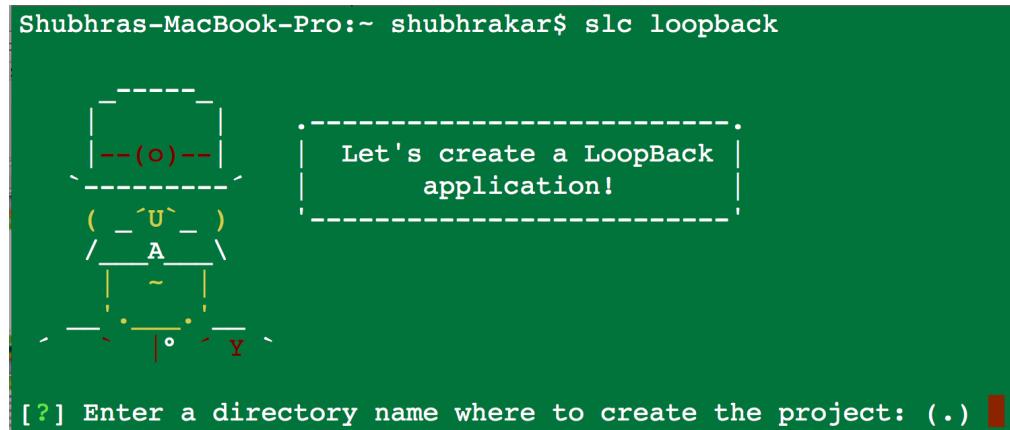
## ■ Arc UI – architect view

- Widget driven
- Visual composition, management & operations
- Marquee featured, evolving platform

The screenshot shows the StrongLoop Arc UI interface. On the left, there's a sidebar with 'MY PROJECT' and sections for 'Models' (Model 01, Model 02, Model 03, Model 04) and 'Datasources' (Oracle 01, mongoDB, apmDev, push). Below these are 'CREATE' buttons for 'New' and 'Render', and icons for various databases: Memory, Oracle, MySQL, Mongo, and Salesforce. The main area is titled 'Model 01' and contains tabs for 'Details', 'Properties', and 'Value'. The 'Properties' tab shows a table with columns: Name, Data Type, Default Value, Required, IsIndex?, and Notes. An example row is 'First Name' with 'String' as the data type. The 'Value' tab has tabs for 'Tab 01' and 'Tab 02'. At the bottom, there's a 'Property Map Editor' with fields for 'First Name', 'Varchar', 'Property Name', and 'String'. A 'Record Collection' section shows 'Array' and '[]'. A vertical 'PROPERTIES' panel is on the far right.

## ■ StrongLoop Console (slc) - developer view

- Command Line Interface
- Full featured
- Greater customization support



# API Explorer (Swagger) – API documentation and agile build

## /locations

Show/Hide | List Operations | Expand Operations | Raw

POST

/locations

Create a new instance of the model and persist it into the data source

PUT

/locations

Update an existing model instance or insert a new one into the data source

GET

/locations/{id}/exists

Check whether a model instance exists in the data source

GET

/locations/{id}

Find a model instance by id from the data source

GET

/locations

Find all instances of the model matched by filter from the data source

GET

/locations/findOne

Find first instance of the model matched by filter from the data source

DELETE

/locations/{id}

Delete a model instance by id from the data source

GET

/locations/count

GET

/locations/nearby

PUT

/locations/{id}

GET

/locations/{id}/inventory



# Data & Services Integration

Model Discovery, Migration &  
Transaction Management !

## NoSQL



{ JSON }



Neo4j



Apache Kafka

A high-throughput distributed messaging system.

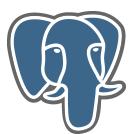


elasticsearch.



ArangoDB

## Database



MySQL

ORACLE

Microsoft  
SQL Server

SAP HANA



## Services

REST



SOAP



SharePoint

&lt;xml/&gt;

twilio  
CLOUD COMMUNICATIONS

atg



## Messaging



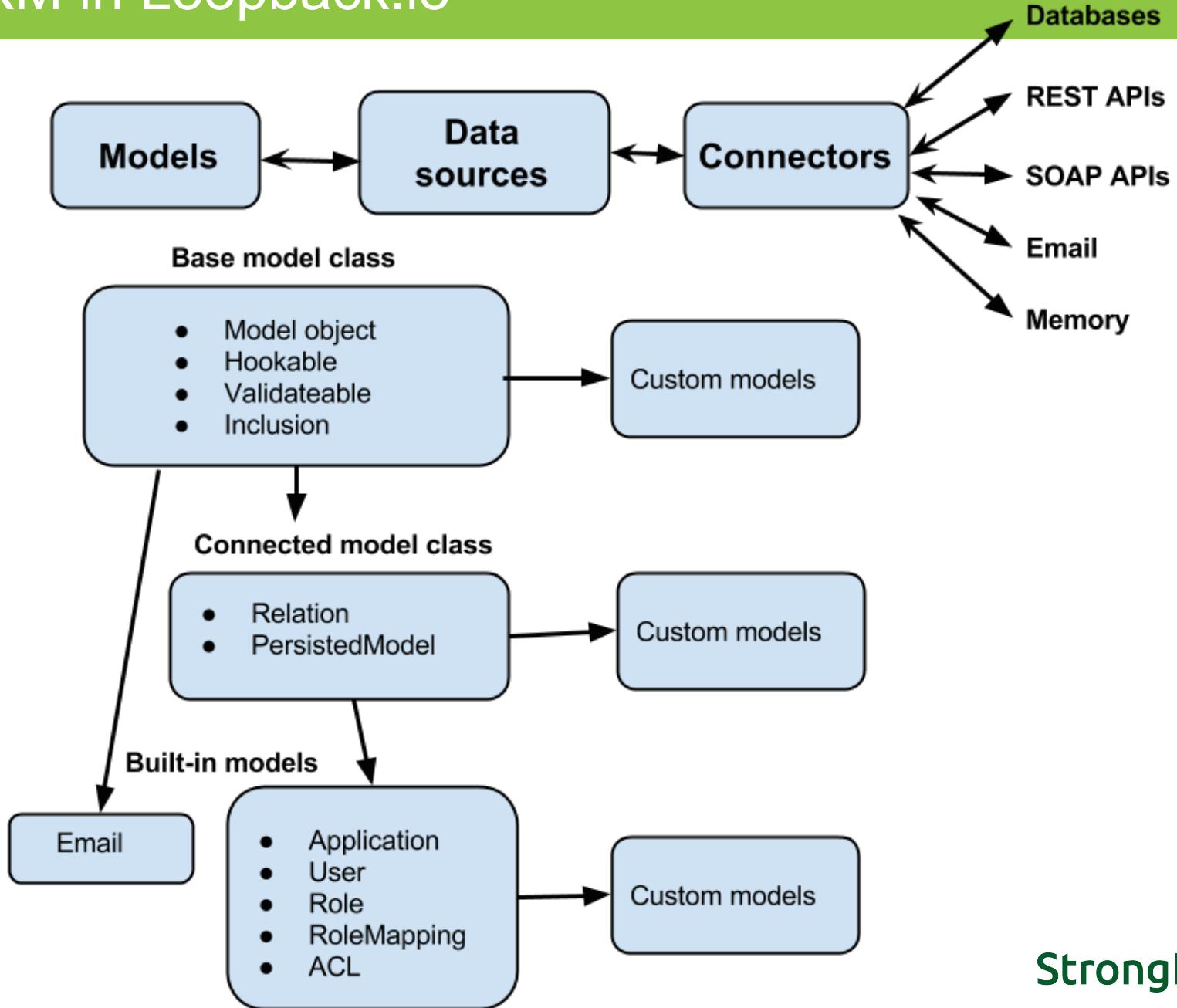
ActiveMQ

WEBSPHERE  
MQ®

redis

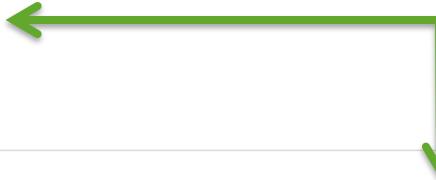
StrongLoop

# ORM in Loopback.io



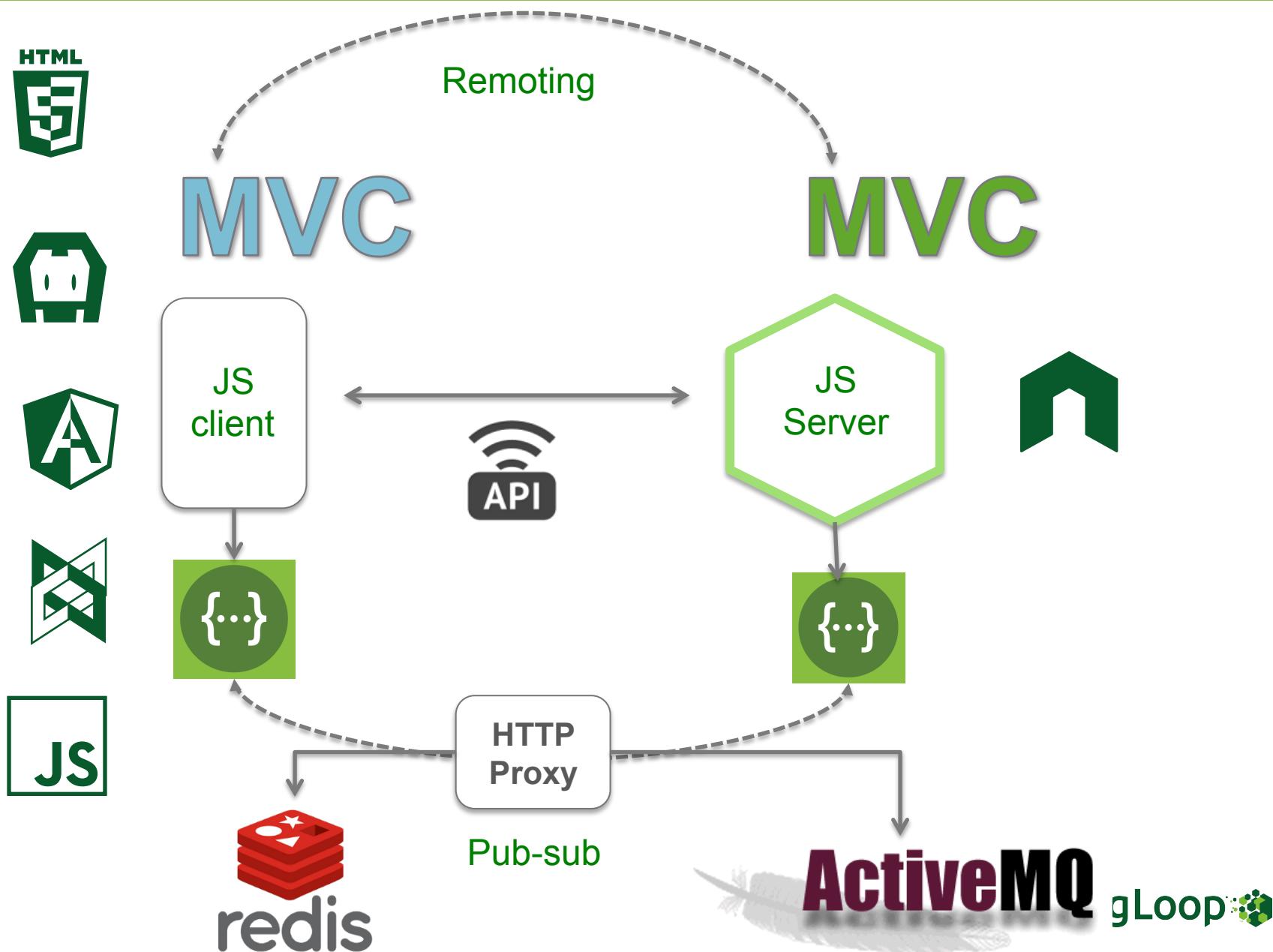
# Service discovery based on API specs

```
{  
  "swagger": "2.0",  
  "info": {  
    "version": "1.0.0",  
    "title": "Swagger Petstore",  
    "description": "A sample API that uses a petstore as an example to demonstrate features in the swagger-2.0 specification",  
    "termsOfService": "http://swagger.io/terms/",  
    "contact": {  
      "name": "Swagger API Team"  
    },  
    "license": {  
      "name": "MIT"  
    }  
  },  
  "host": "petstore.swagger.io",  
  "basePath": "/api",  
  "schemes": [  
    "http"  
  ],  
  "consumes": [  
    "application/json"  
  ],  
  "produces": [  
    "application/json"  
  ],  
  "paths": {  
    "/pets": {  
      "get": {  
        "description": "Ret  
        "operationId": "fir  
        "produces": [  
          "application/json"  
          "application/xml"  
          "+text/xml".  
      }  
    }  
  }  
}
```

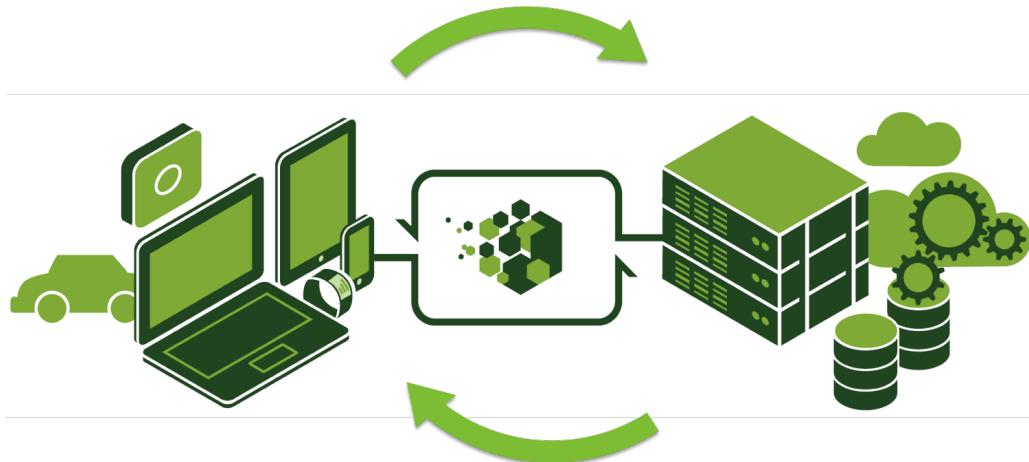


PetInputs		
POST	/PetInputs	Create a new instance of the model and persist it into the data source.
GET	/PetInputs	Find all instances of the model matched by filter from the data source.
PUT	/PetInputs	Update an existing model instance or insert a new one into the data source.
PUT	/PetInputs/{id}	Update attributes for a model instance and persist it into the data source.
HEAD	/PetInputs/{id}	Check whether a model instance exists in the data source.
GET	/PetInputs/{id}	Find a model instance by id from the data source.
DELETE	/PetInputs/{id}	Delete a model instance by id from the data source.
GET	/PetInputs/{id}/exists	Check whether a model instance exists in the data source.
GET	/PetInputs/count	Count instances of the model matched by where from the data source.
GET	/PetInputs/findOne	Find first instance of the model matched by filter from the data source.
POST	/PetInputs/update	Update instances of the model matched by where from the data source.

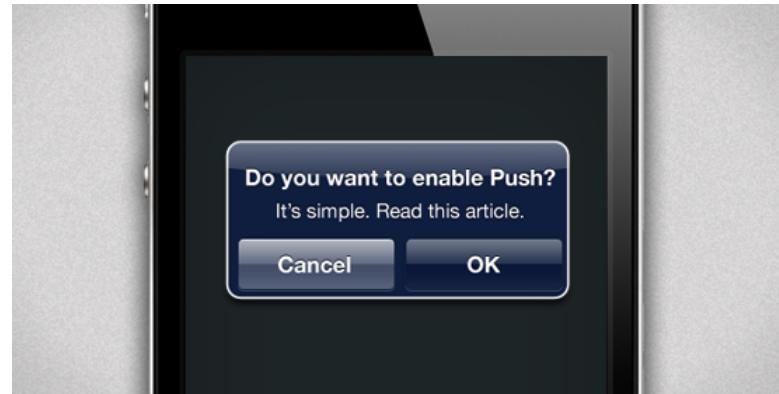
# Isomorphic JS in Loopback.io



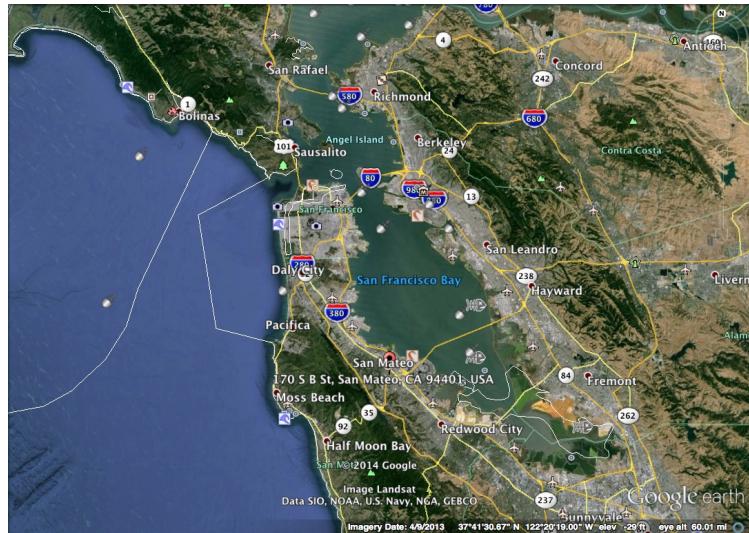
# Mobile Backend Services



Offline Sync



Push Notification



Geo Location



Storage

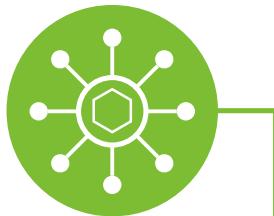
**StrongLoop**

# A node.js Gateway is coming up too !

Security & Social Logins



## API Clients



CLIENT  
APPLICATION



API

## API Gateway



Authorization  
Endpoint

Token Endpoint

oAuth 2.0  
Bearer/JWT  
Middleware

Rate  
Limiting

HTTP  
Proxy

## API Server



PROTECTED  
API  
ENDPOINTS

StrongLoop

# StrongLoop – node.js Development to Production

## StrongLoop Arc



**API Composer**  
Visual modeling



**Profiler**  
Root cause  
CPU & Memory



**Performance Metrics**  
Real-time  
production  
monitoring



**Build and Deploy**  
Automate  
Lifecycle



**Process Manager**  
Scale  
applications



express



**LoopBack**

ORM, mBaaS, Realtime

**StrongLoop**