

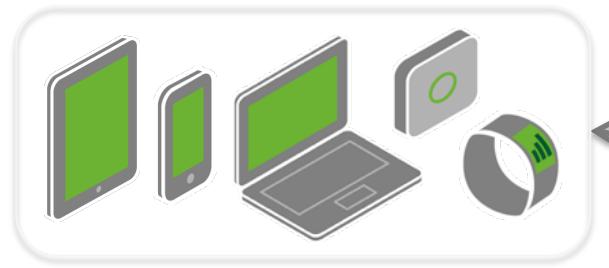
Wake up to the API economy

Explosion of channels
(browsers, mobile, sensors,
wearable, real-time
communication)

- ◆ Sheer number of clients has increased
- ◆ Clients are increasingly sophisticated
- ◆ Node.js is the platform of choice for these API



Not all APIs are built equal



X - Channel



X - Platform

API Neutrality



X – Backend ?

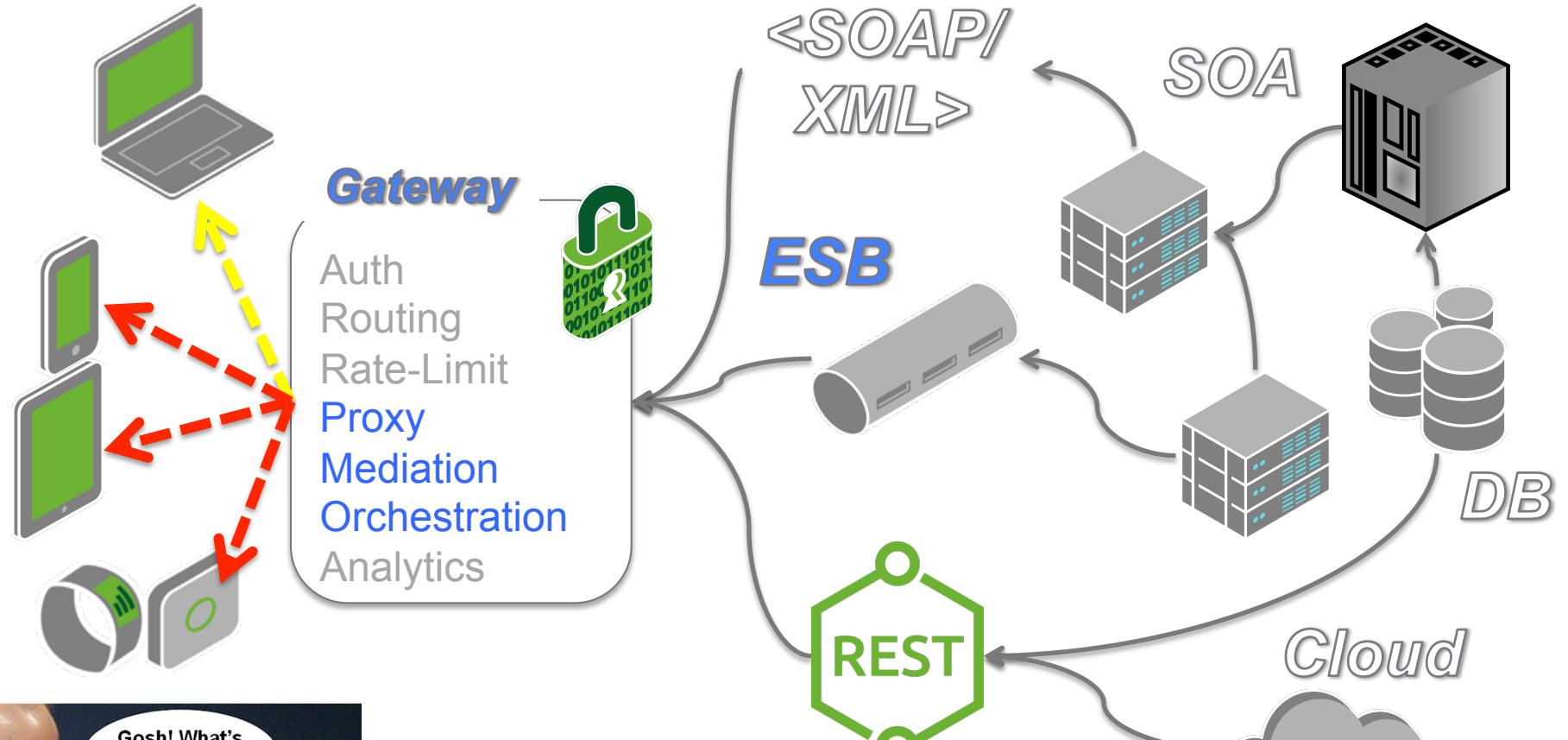


Mainframe to Mobile ?

StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop © 2013

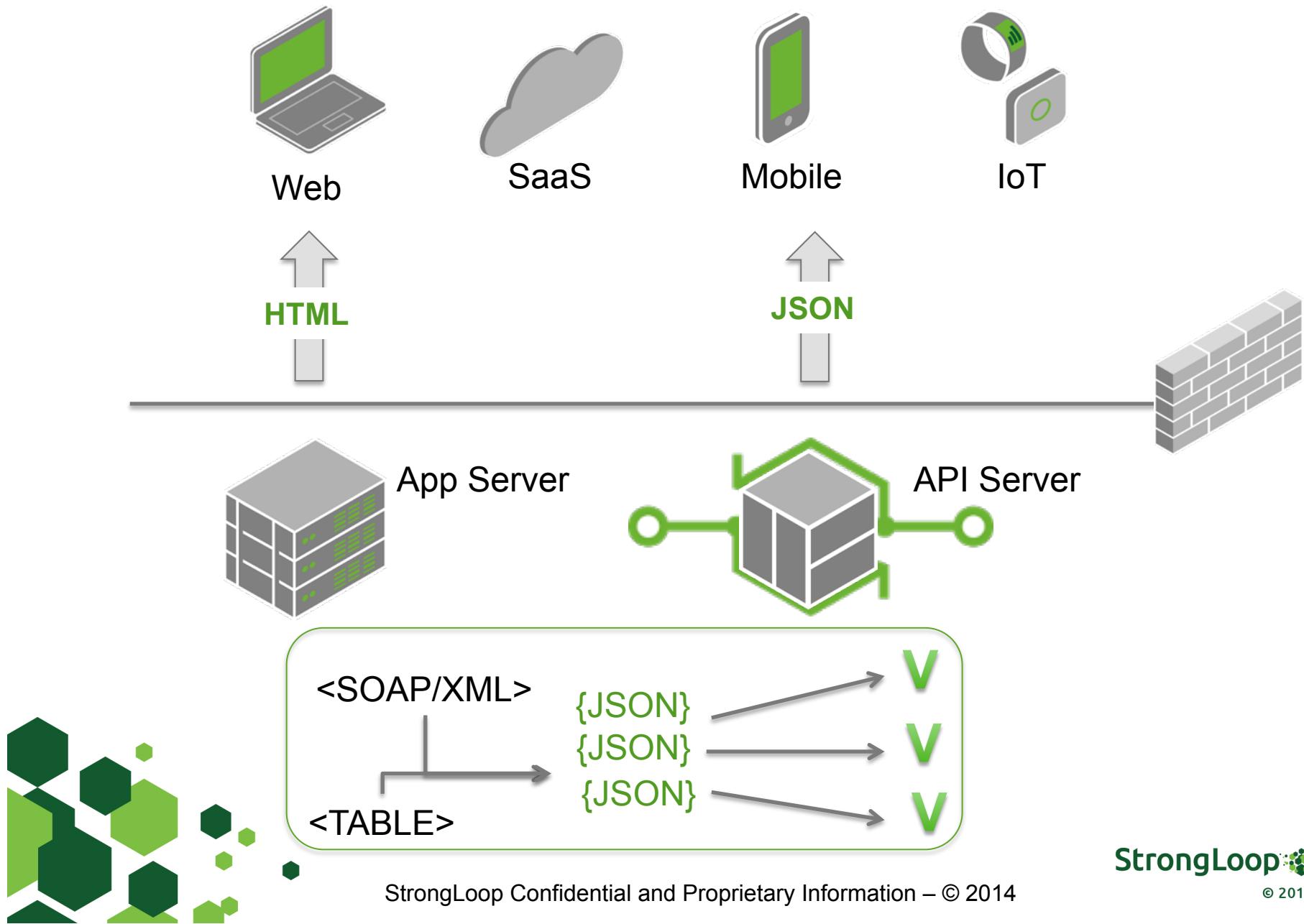
API “Re-Composition” is good...but not good enough



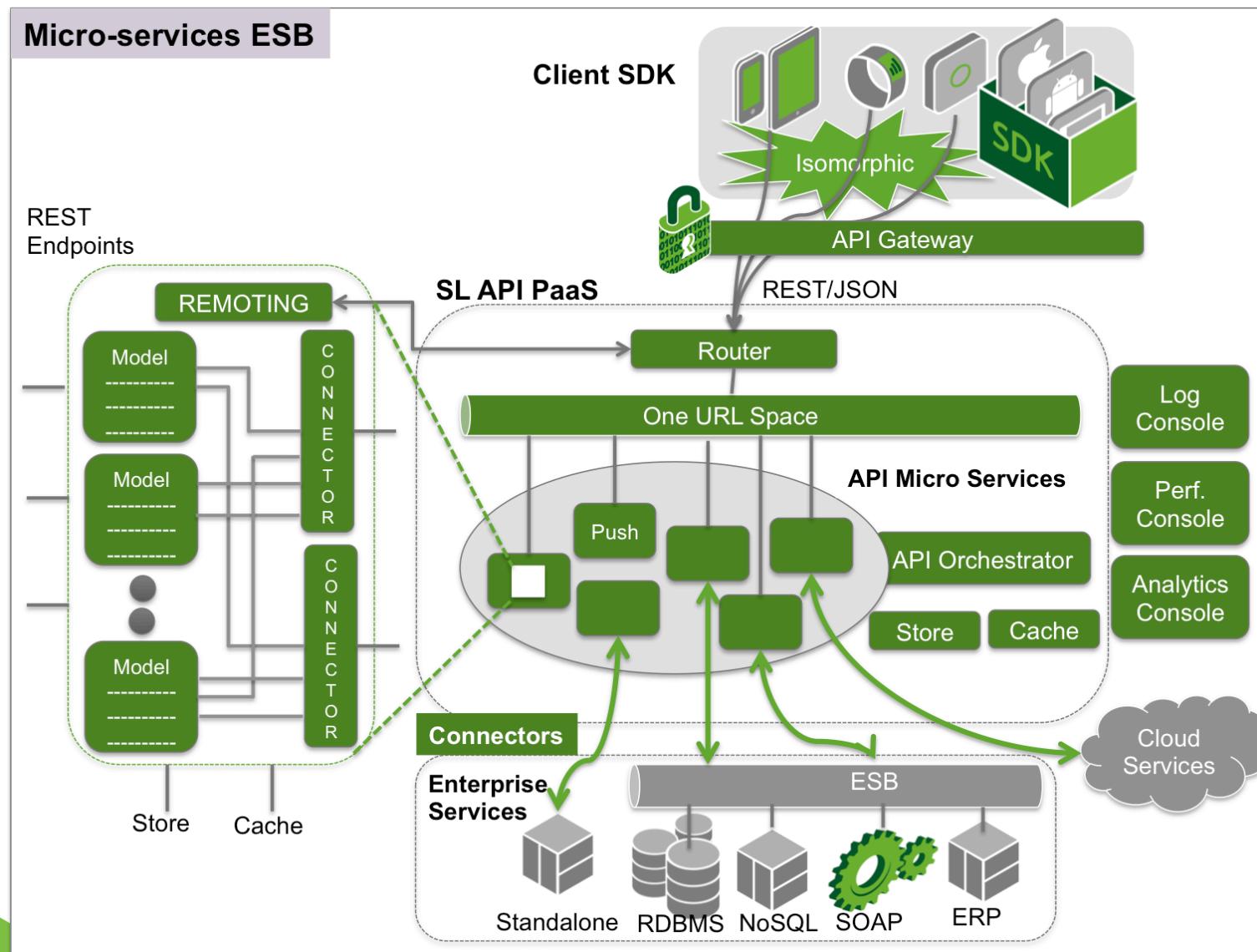
StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop 
© 2013

The need is “DE-COMPOSITION”



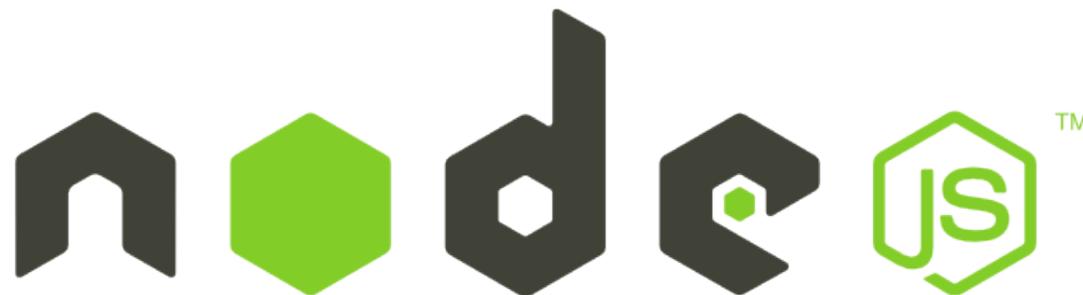
Micro-services has arrived



Now I know !!!



So who is propelling this transformation ?



Why Node?

Node is *FAST...er*

...and highly concurrent!

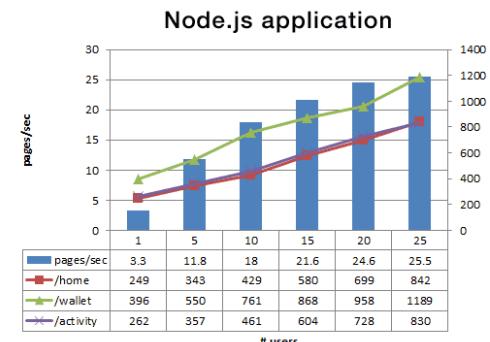
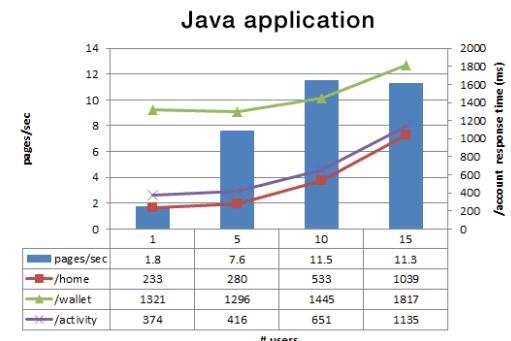
Node is much lighter

Node is perfect for APIs

Node is JavaScript & JSON



StrongLoop Confidential and Proprietary Information – © 2015



Any use-cases ?

- ◆ **LinkedIn moved from Rails to Node.js**

- Cut down servers by 90% ($30 \rightarrow 3$)
 - Lower memory overhead and up to 20X faster

- ◆ **PayPal migrated to Node.js from Java:**

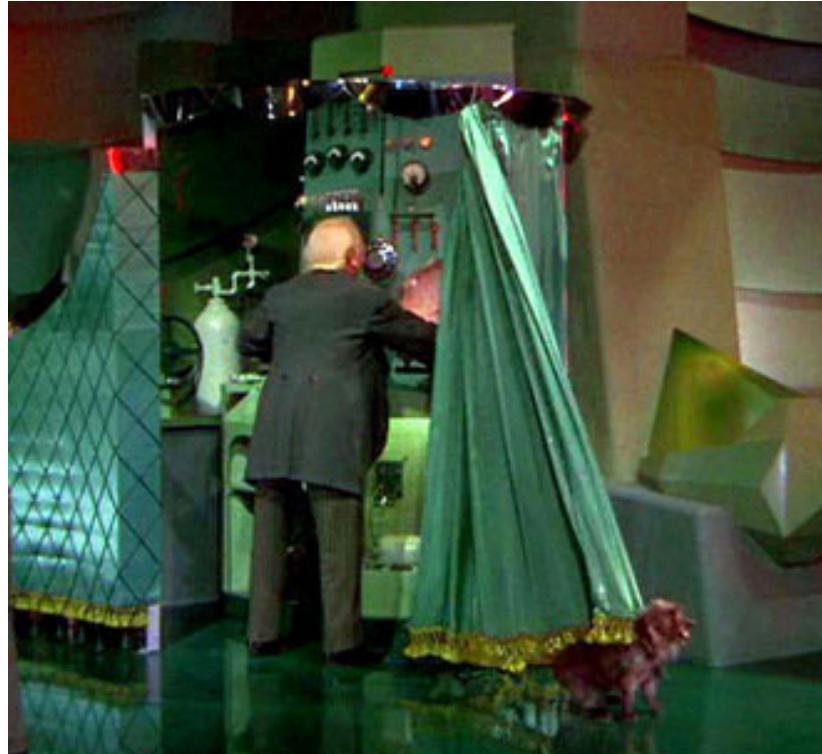
- Reduce development time by >50%
 - 2X speed of delivery with less than existing resources
 - 33% lesser lines of code
 - 40% fewer files

- ◆ **Groupon displaced Ruby and Java with Node.js across the board:**

- 50% improvement in page response times
 - Agile builds and deploys



Behind the Curtain



Yes, it's C back there

Node Standard Library (JavaScript)

Node Bindings C/C++
(socket, http, etc.)

V8

Libuv

Libeio
(thread pool)

~~Libev~~
~~(event loop)~~

IOCP
(IO completion ports)

epoll/kqueue/poll (event ports)

Windows Kernel

Linux Kernel

{".."} JSON is the new first class citizen

Single Page Apps



Server Side JavaScript
(Node.js)



JSON REST API

JSON Document DB



m e A n

Full Stack JavaScript is powering mobility



Gartner predicts by 2016 more than 50% of the apps deployed will be Hybrid apps.



StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop
© 2013

Nodies are not just Silicon Valley hipsters !

NODE IS DEPLOYED BY BIG BRANDS

Big brands are using Node to power their business

Manufacturing



General Motors



SIEMENS

Financial



Goldman
Sachs

PayPal



eCommerce

amazon.com



ebay



Zappos[®].com

Media



CONDÉ NAST

DOWJONES

The New York Times

SONY

Technology

salesforce.com



box

intel

YAHOO!

IBM[®]



airbnb

PEARSON

STAPLES[®]



HBO[®]

BARNES&NOBLE



redhat.



NETFLIX

npr

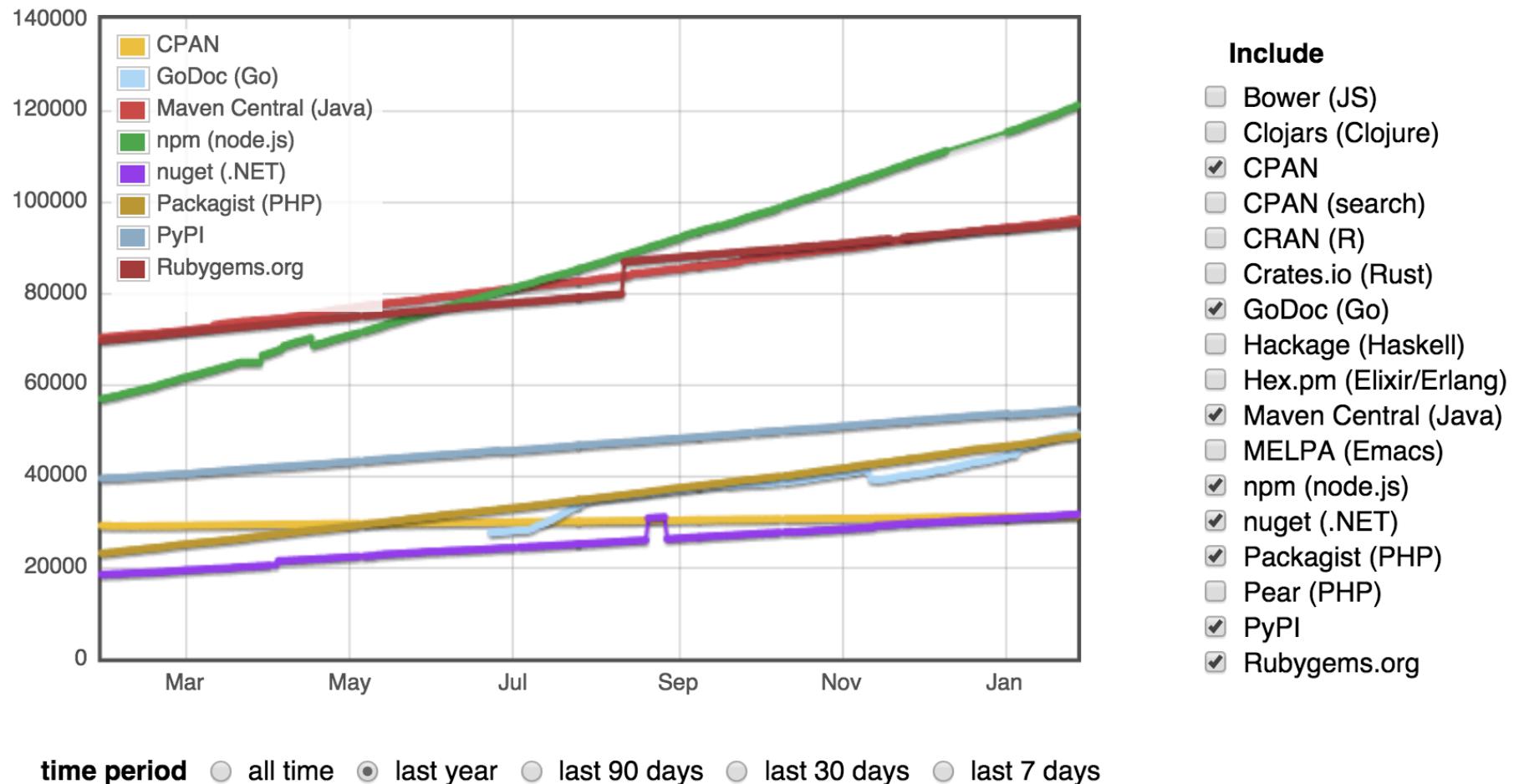
StrongLoop[®]

© 2013

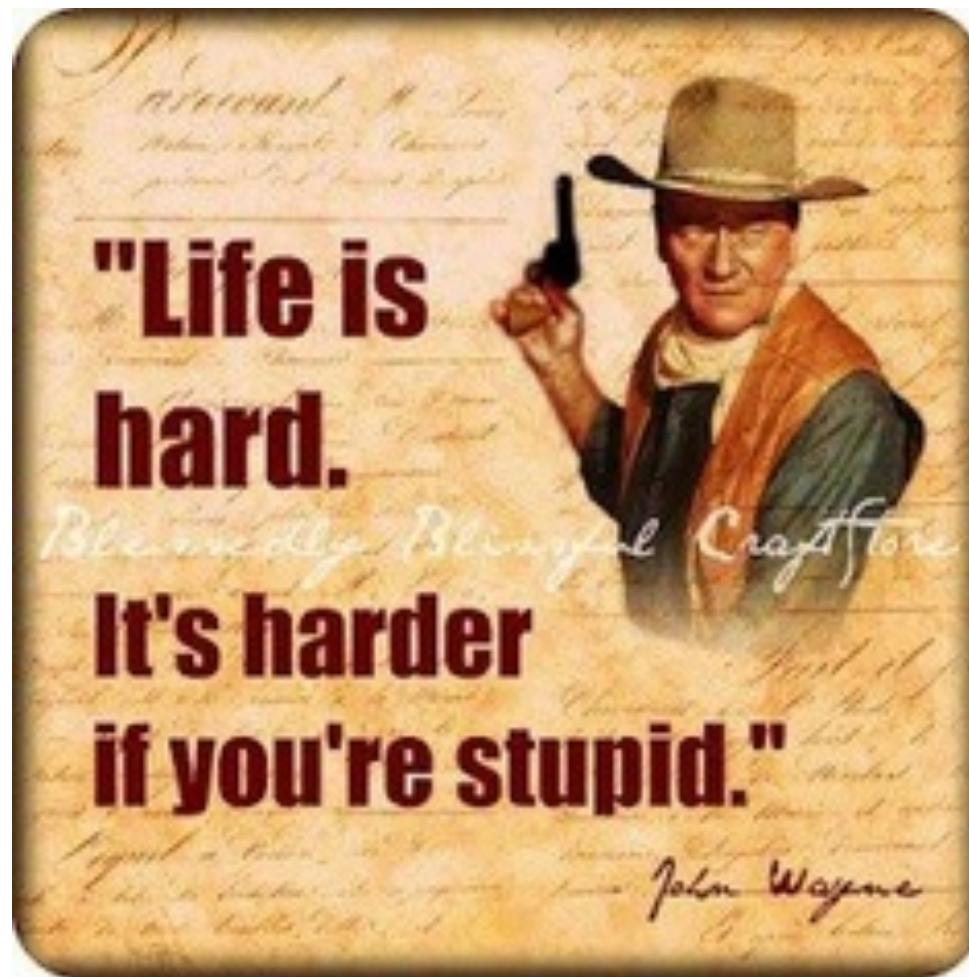
Node is de-facto for the backend API now

- 67K Developers
- Fastest Growing (120K+ > Java, Ruby, PHP...)

Module Counts



Really, But isn't it the wild west ?





StrongLoop™

Node Core



Ben
Noordhuis



Bert
Belder



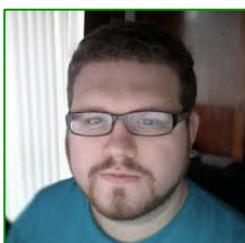
Miroslav
Bajtos



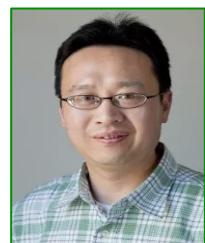
Sam
Roberts



Ryan
Graham



Ritchie
Martori



Raymond
Feng

LoopBack / Express Core

5 of the top 10 Node Contributors

StrongLoop
developers made

• 420+ Commits

to Node v0.12

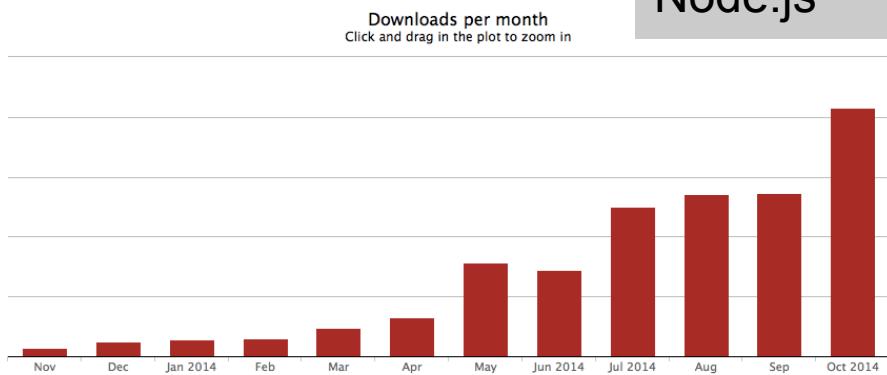
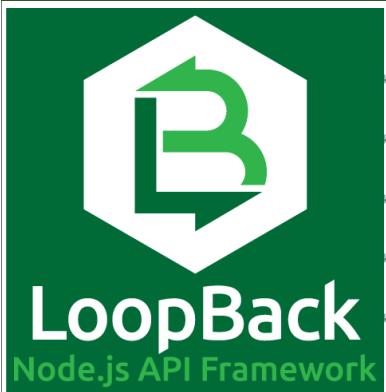
1	Ben Noordhuis
2	Fedor Indutny
3	Isaac Schlueter
4	Timothy Fontaine
5	Trevor Norris
6	Bert Belder
7	Domenic Denicola
8	Miroslav Bajtos
9	Sam Roberts
10	Ryan Graham

Github: March 28, 2013 - July 15, 2014

Plus 15 other full-time open source developers working on Node.js core, modules and tools.

Contributing to over 100 open source modules.

Supporting banks, retail, IoT companies, startups and internet companies on Node.js



Express

The only product company dedicated only to Node



Jason Pressman



Kiran Prasad



Nick Sturiale

ignition
splunk>



Neil Day



Marten Mickos



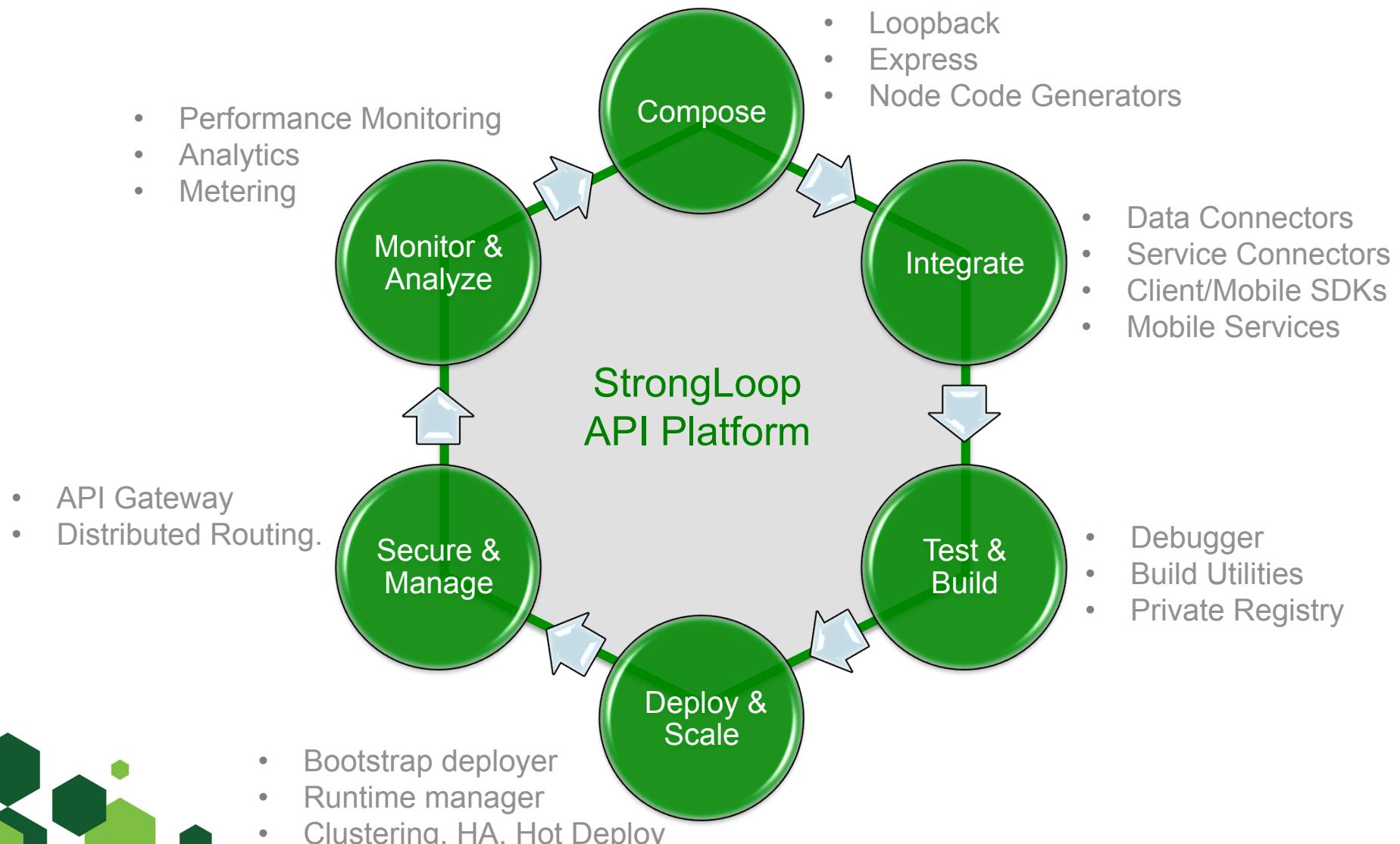
Paul Ambrose



WebLogic

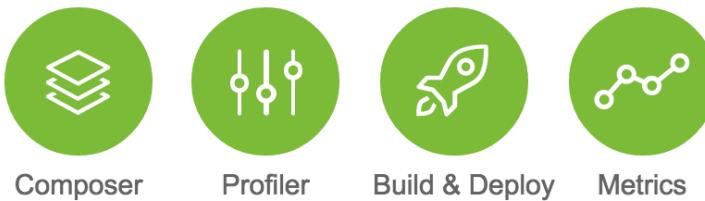


StrongLoop iPaaS for APIs



Introducing StrongLoop Arc

A unified graphical interface with tools for the API lifecycle



The screenshot shows the StrongLoop Arc interface with three main tabs:

- Composer**: Shows a navigation tree for "ARC-GETTING-STARTED" with sections like "Models" (selected), "Review", and "DataSources". It includes a "Module picker" button, a "Main model / data source editing view" for "CoffeeShop", and a "Models & data sources creation" section.
- Profiler**: Displays a CPU profile for "node_1322" with the title "Heavy (Bottom Up)". The table lists various functions and their execution times. A green arrow points from the "Build & Deploy" tab towards this section.
- Build & Deploy**: Shows deployment options like "Deploy tar file" (existing archive) and "Deploy" (with fields for Hostname, Port, and Processes).

Popular Frameworks



express

High performance, high class web development for **Node.js**

[Home](#) [Guide](#) [Screencasts](#) [Applications](#)

```
var app = express.createServer();
```



Sample Code (Router using express 4.x)

```
var express = require('express');
var Item = require('models').Item;
var app = express();
var itemRoute = express.Router();

itemRoute.param('itemId', function(req, res, next, id) {
  Item.findById(req.params.itemId, function(err, item) {
    req.item = item;
    next();
  });
});

itemRoute.route('/:itemId')
  .get(function(req, res, next) {
    res.json(req.item);
  })
  .put(function(req, res, next) {
    req.item.set(req.body);
    req.item.save(function(err, item) {
      res.json(item);
    });
  })
  .post(function(req, res, next) {
    var item = new Item(req.body);
    item.save(function(err, item) {
      res.json(item);
    });
  })
  .delete(function(req, res, next) {
    req.item.remove(function(err) {
      res.json({}); // empty object
    });
  });
;

app.use('/api/items', itemRoute);
app.listen(8080);
```

Pros

- Great starting point
- Low learning curve
- Nearly a standard for Node.js web middleware
- Fully customizable
- Targeted at browser, hence templating and rendering OOB



Cons

- All endpoints need to be created manually
- Developers may end up creating their own non-optimized libraries
- Every end point needs to be tested
- Refactoring is painful, update everything everywhere
- No standards or prescription, DIY



restify



Sample Code (Router using restify)

```
var restify = require('restify');
var Item = require('models').Item;
var app = restify.createServer()

app.use(function(req, res, next) {
  if (req.params.itemId) {
    Item.findById(req.params.itemId, function(err, item) {
      req.item = item;
      next();
    });
  } else {
    next();
  }
});

app.get('/api/items/:itemId', function(req, res, next) {
  res.send(200, req.item);
});

app.put('/api/items/:itemId', function(req, res, next) {
  req.item.set(req.body);
  req.item.save(function(err, item) {
    res.send(204, item);
  });
});

app.post('/api/items/:itemId', function(req, res, next) {
  var item = new Item(req.body);
  item.save(function(err, item) {
    res.send(201, item);
  });
});

app.delete('/api/items/:itemId', function(req, res, next) {
  req.item.remove(function(err) {
    res.send(204, {});
  });
});

app.listen(8080);
```

Pros



Very stable and actively developed.

Automatic DTrace support for all handlers on enabled platforms

Built in throttling

SPDY support

Borrows from express but without unnecessary templating/rendering

Strict API with full control over HTTP interactions

Visibility into latency and app characteristics

Cons



All endpoints need to be created manually

Developers may end up creating their own non-optimized libraries

Every end point needs to be tested

Refactoring is painful, update everything everywhere

No standards or prescription, DIY

Less documentation





StrongLoop
© 2013

Sample Code (Router using hapi)

```
var Hapi = require('hapi');
var Item = require('models').Item;
var server = Hapi.createServer('0.0.0.0', 8080);

server.ext('onPreHandler', function(req, next) {
  if (req.params.itemId) {
    Item.findById(req.params.itemId, function(err, item) {
      req.item = item;
      next();
    });
  } else {
    next();
  }
});

server.route([
  {
    path: '/api/items/{itemId}',
    method: 'GET',
    config: {
      handler: function(req, reply) {
        reply(req.item);
      }
    }
  },
  {
    path: '/api/items',
    method: 'PUT',
    config: {
      handler: function(req, reply) {
        req.item.set(req.body);
        req.item.save(function(err, item) {
          res.send(204, item);
        });
      }
    }
  }
]);
```

Sample Code (Router using hapi)

```
{  
  path: '/api/items',  
  method: 'POST',  
  config: {  
    handler: function(req, reply) {  
      var item = new Item(req.body);  
      item.save(function(err, item) {  
        res.send(201, item);  
      });  
    }  
  }  
},  
{  
  path: '/api/items/{itemId}',  
  method: 'DELETE',  
  config: {  
    handler: function(req, reply) {  
      req.item.remove(function(err) {  
        res.send(204, {});  
      });  
    }  
  }  
}  
]);  
  
server.start();
```



Pros

- Supported by Walmart labs. Less known but getting momentum
- Very granular control over request handling
- Detailed API reference with support for document generation
- Significantly more functionality for building web servers
- Pseudo prescriptive with configuration centric approach
- Caching, Authentication and input validation available
- Plugin based architecture for scaling

Cons

- Good construction blocks but left to your own devices to figure out design patterns
- Binds you to specific modules and tool chains (catbox, joi, tv, boom, good, travelogue, yar)
- All endpoints need to be created manually
- Developers may end up creating their own non-optimized libraries
- Every end point needs to be tested
- Refactoring is painful, update everything everywhere





StrongLoop
© 2013

Sample Code (Router using Loopback)

Code needed

```
var loopback = require('loopback');
var Item = require('./models').Item;
var app = module.exports = loopback();

app.model(Item);
app.use('/api', loopback.rest());
app.listen(8080);
```

Endpoints automatically created

```
DELETE /items/{id}
GET /items
GET /items/count
GET /items/findOne
GET /items/{id}
GET /items/{id}/exists
POST /items
PUT /items
PUT /items/{id}
```



Sample Code (API explorer and test)

Code needed

```
var explorer = require('loopback-explorer');
app.use('/explorer', explorer(app, {basePath: '/api'}));
```

API explorer automatically created



/items

Show/Hide | List Operations | Expand Operations | Raw

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

[BASE URL: <http://0.0.0.0:8080/api>]

Sample Code (RPC)

Code needed

```
var loopback = require('loopback');
var explorer = require('loopback-explorer');
var remoting = require('strong-remoting');
var Item = require('./models').Item;
var app = module.exports = loopback();
var rpc = remoting.create();

function echo(ping, callback) {
  callback(null, ping);
}

echo.shared = true;
echo.accepts = {arg: 'ping'};
echo.returns = {arg: 'echo'};

rpc.exports.system = {
  echo: echo
};

app.model(Item);
app.use('/api', loopback.rest());
app.use('/explorer', explorer(app, {basePath: '/api'}));
app.use('/rpc', rpc.handler('rest'));
app.listen(8080);
```

Remotable calls

```
$ curl "http://localhost:8080/rpc/system/echo?ping=hello"
{
  "echo": "hello"
}
```

Pros

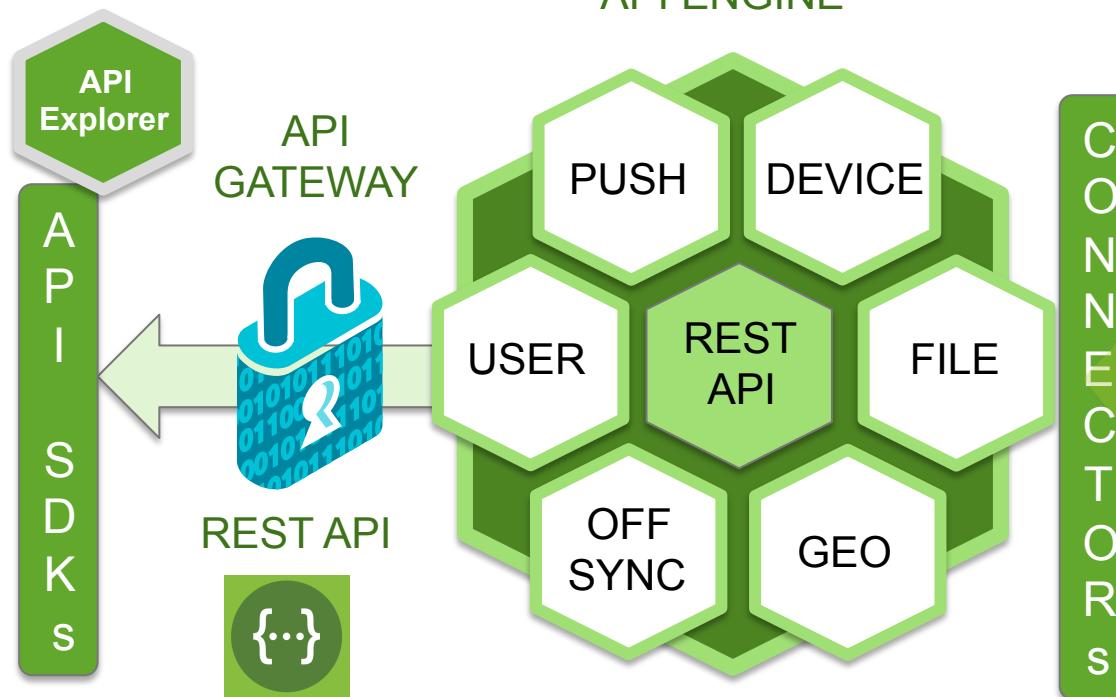
- Very quick RESTful API development
- Convention over configuration (extends express)
- Built in models ready to use (ORM)
- RPC support, Web and Mobile SDKs
- Data Connectors
- Fully configurable when needed
- Extensive documentation
- Commercial support and full time team working on project (Loopback)
- Mobile Services (Push, Geo, Storage, Sync, Offline, File, Device, User) - Loopback
- Authentication, validation and Authorization

Cons

- Higher learning curve
- Opinionated

Loopback framework – ORM and Mobile connect in Node.js

Channel SDKs



Magical Features

- ◆ Discovery
- ◆ Migration / Replication
- ◆ Relationship Mapping
- ◆ Mixins - Aggregation
- ◆ Access Control
- ◆ User Management
- ◆ Isomorphic JS / Remoting
- ◆ Routing



StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop 
© 2013

Data & Services Integration

Marquee - Discovery & Migration !

NoSQL



Database



Services



Messaging



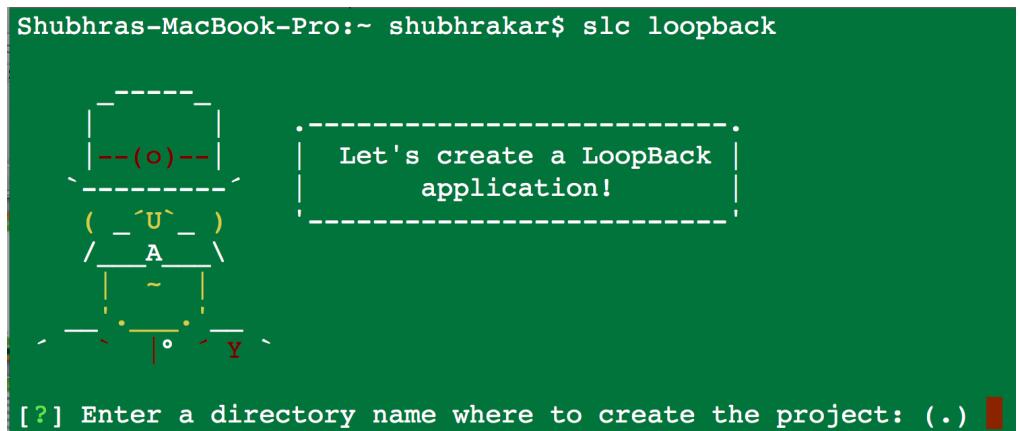
User Experience

◆ Arc UI – architect view

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

◆ StrongLoop Console (slc) - developer view

- Command Line Interface
- Full featured
- Greater customization support

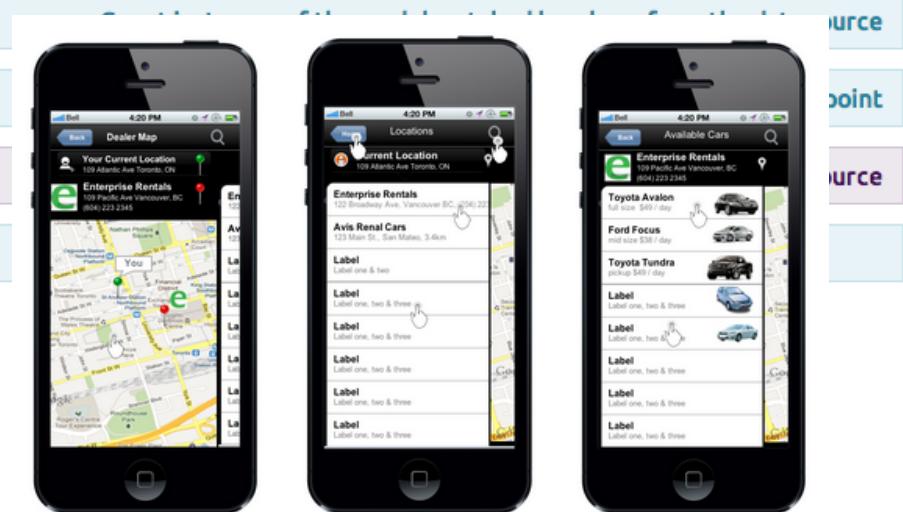


API Explorer (Swagger) – Iterative build and test w/o frontend

/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	



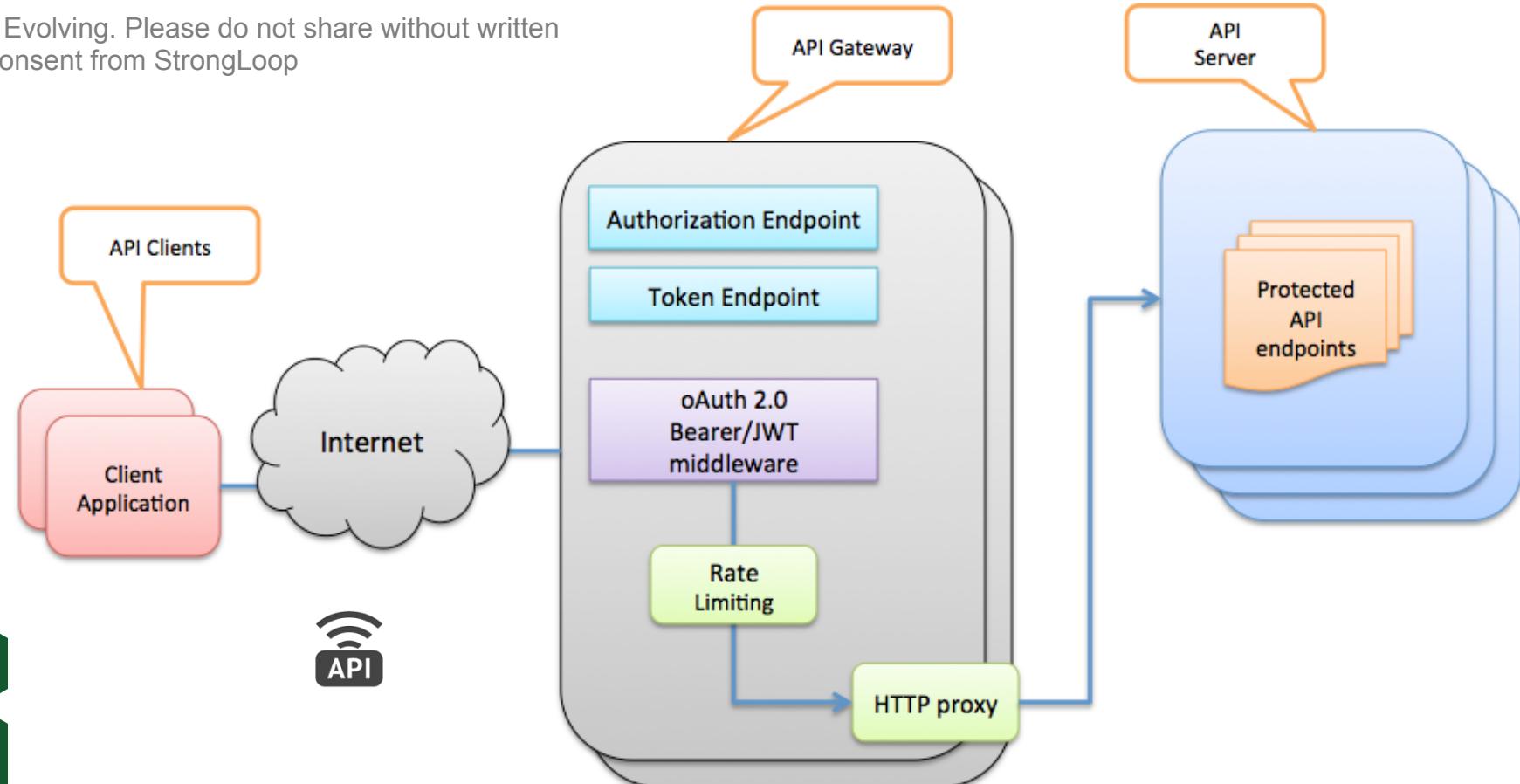
API Gateway *

Marquee – Auth, Routing, Throttling, Proxy, Mediation, Aggregation, Virtualization

Security & Social Logins



* Evolving. Please do not share without written consent from StrongLoop



API Gateway*

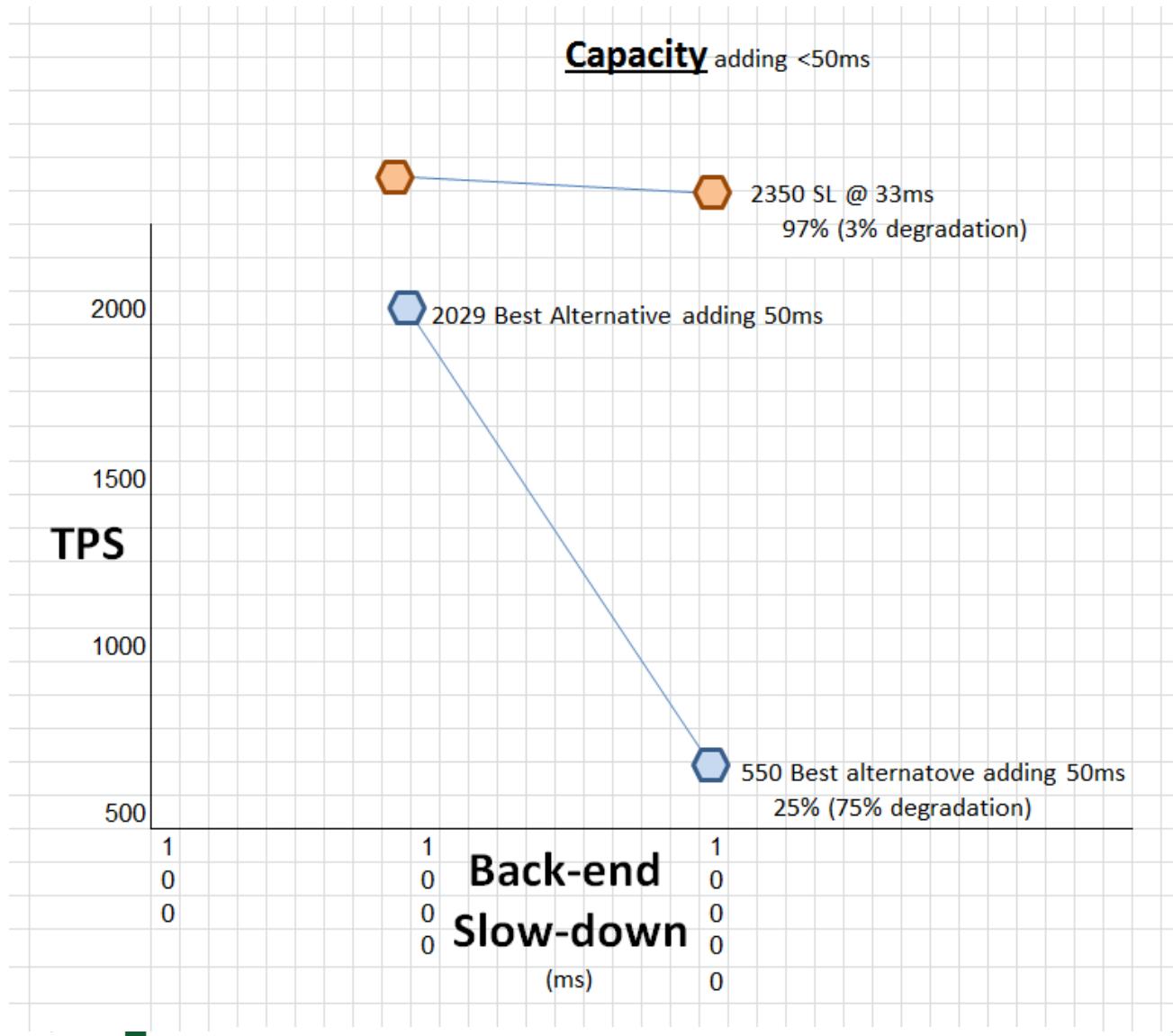
- ◆ Security (AAA)
- ◆ Mediation and Transformation
- ◆ Infrastructure QoS (pagination, throttling, caching, delivery guarantee, firewall)
- ◆ Analytics and health monitoring
- ◆ Aggregation and Microservices Routing
- ◆ Virtualization and Reverse Proxy



* Evolving. Please do not share without written consent from StrongLoop

Node vs Java for REST routing *

* Highly Confidential. Please do not share without written consent from StrongLoop



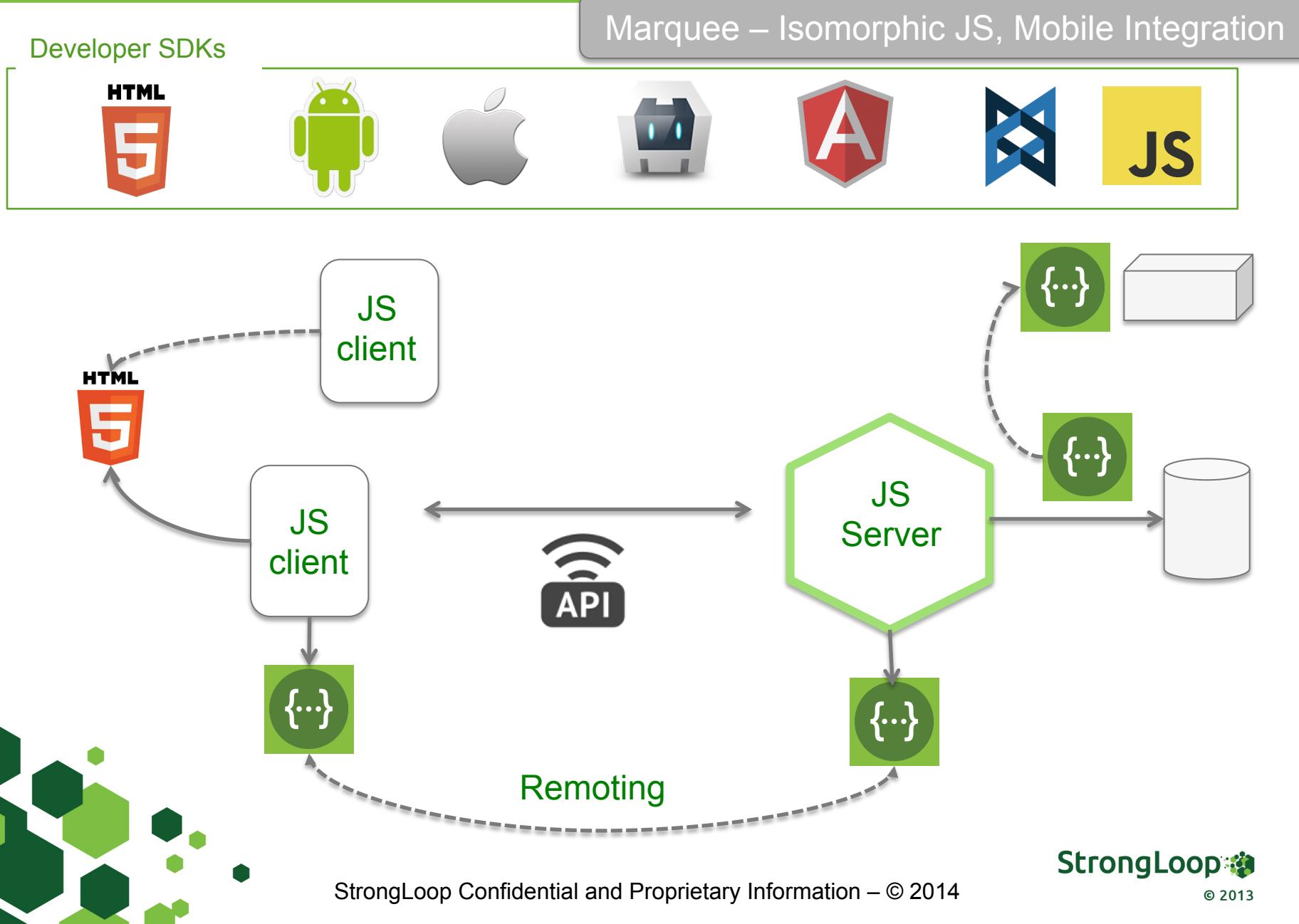
Some Numbers – Top 3 US Bank *

* Highly Confidential. Please do not share without written consent from StrongLoop

Sunny Day = 1 sec backend delay Rainy Day = 10 sec backend delay Test Duration = 1 hour		Performance Baseline Results							
		Latency Target = <u>100 ms</u>				Latency Target = <u>50 ms</u>			
		Users	Ramp-up (seconds)	Latency Observed (ms)	TPS per Server	Users	Ramp-up (seconds)	Latency Observed (ms)	TPS per Server
Mule	Sunny Day	2000	60	90	1819	1000	30	50	951
	Rainy Day	6000	120	77	368	4150	120	40	311
Strong Loop	Sunny Day	3189	1	93.5	2756*	3000	1000	41	2792
	Rainy Day	16000	1000	55.3	2492	18000	1000	26	2350



Omni-channel Client SDKs

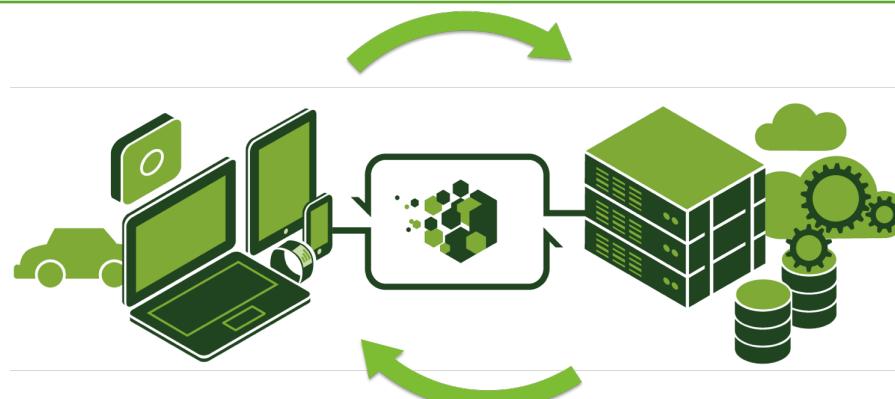


Mobile Services

Marquee – Push, Geo, Offline Sync, Storage



mBaaS Capabilities



LoopBack Storage Service

Select files

Base drop zone

Another drop zone with its own settings

Multiple Choose Files No file chosen

Single Choose File No file chosen

Upload queue

Queue length: 0

Name	Size	Progress	Status	Actions

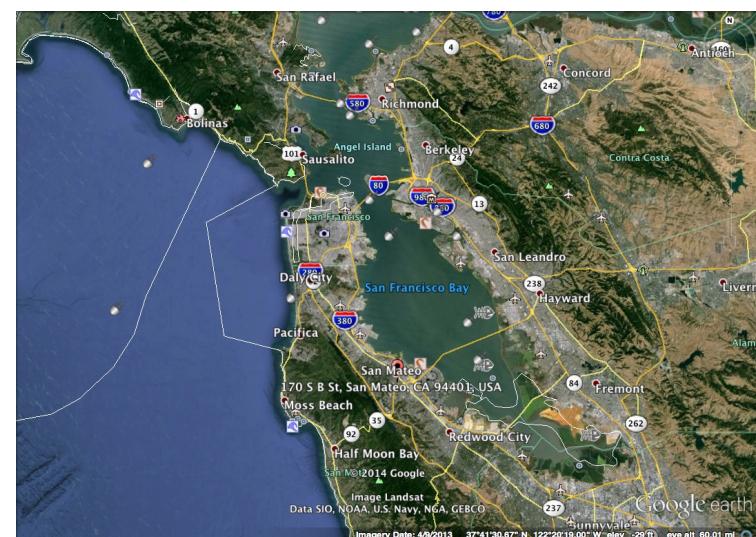
Queue progress:

Upload all Cancel all Remove all

Files in the container

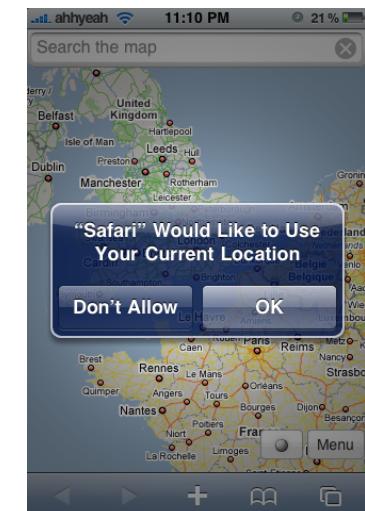
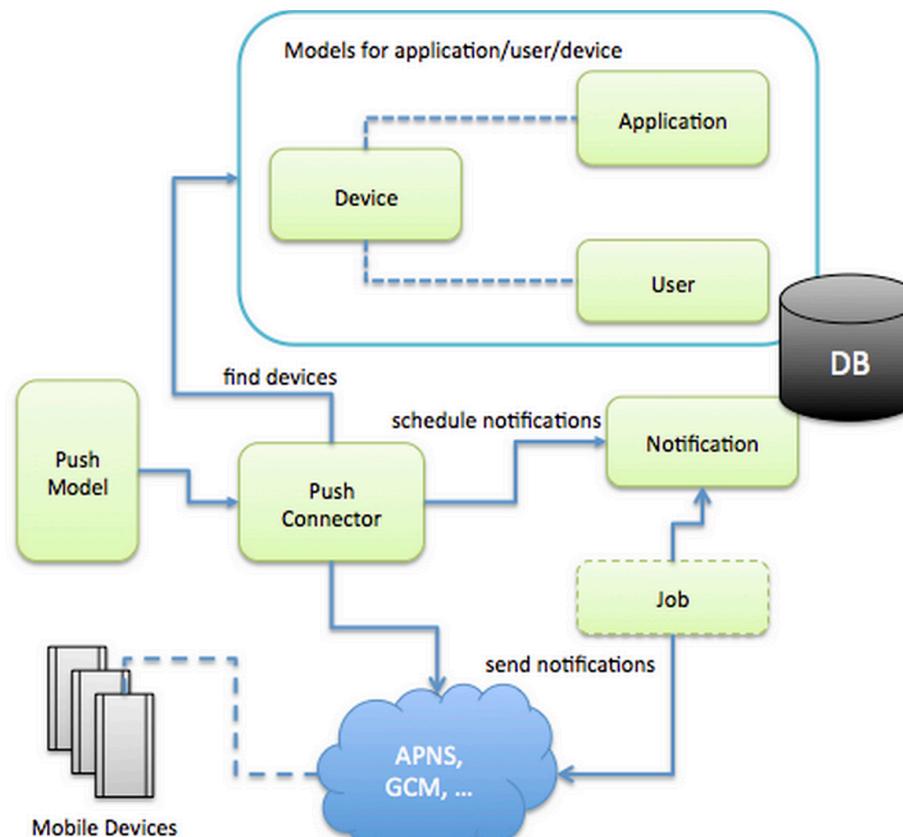
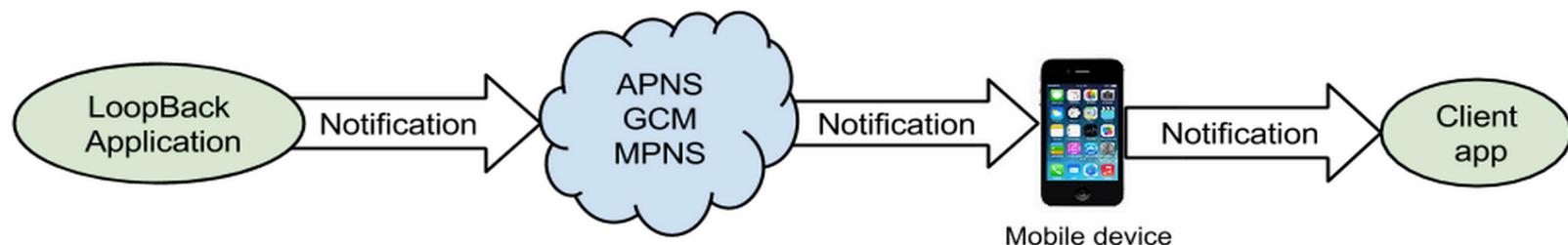
f1.txt Remove

f1_downloaded.txt Remove



StrongLoop Confidential and Proprietary Information – © 2014

Push Notification



GeoLocation

```
...
  "geoRest": {
    "connector": "rest",
    "debug": "false",
    "operations": [
      {
        "template": {
          "method": "GET",
          "url": "http://maps.googleapis.com/maps/api/geocode/{format=json}",
          "headers": {
            "accepts": "application/json",
            "content-type": "application/json"
          },
          "query": {
            "address": "{street},{city},{zipcode}",
            "sensor": "{sensor=false}"
          },
          "responsePath": "$.results[0].geometry.location"
        },
        "functions": {
          "geocode": [ "street", "city", "zipcode"]
        }
      }
    ]
  }
...
}
```



```
CoffeeShop.attachTo(oracle);
var here = new GeoPoint({lat: 10.32424, lng: 5.84978});
CoffeeShop.find( {where: {location: {near: here}}}, limit:3, function(err, nearbyShops) {
  console.info(nearbyShops); // [CoffeeShop, ...]
});
```

Storage Service

server/server.js

```
var ds = loopback.createDataSource({  
  connector: require('loopback-component-storage'),  
  provider: 'amazon',  
  key: 'your amazon key',  
  keyId: 'your amazon key id'  
});
```

```
var container  
app.model(con
```

LoopBack Storage Service

Select files

Base drop zone

Another drop zone with its own settings

Multiple
 No file chosen

Single
 No file chosen

Upload queue

Queue length: 0

Name	Size	Progress	Status	Actions
------	------	----------	--------	---------

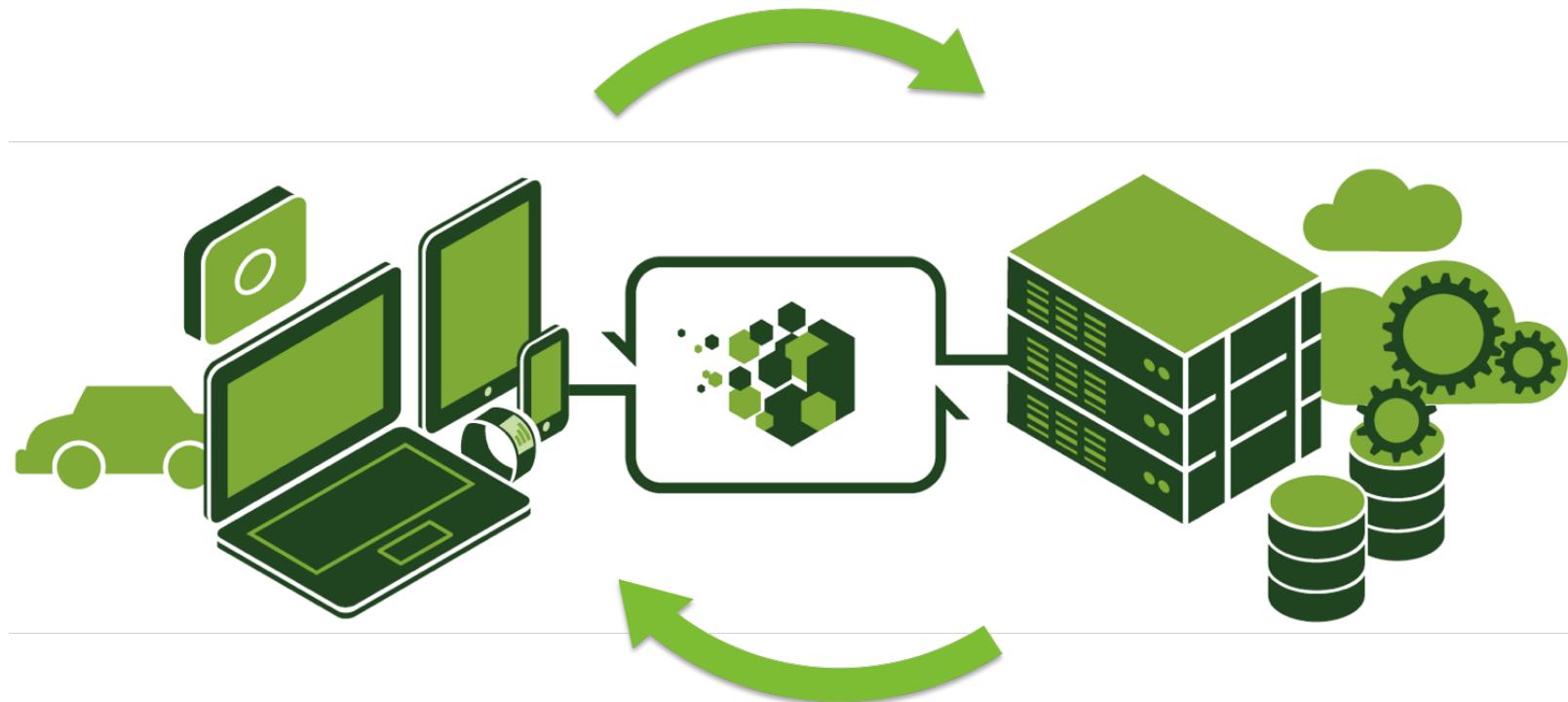
Queue progress:

Files in the container

f1.txt	<input type="button" value="Remove"/>
f1_downloaded.txt	<input type="button" value="Remove"/>



Offline Sync



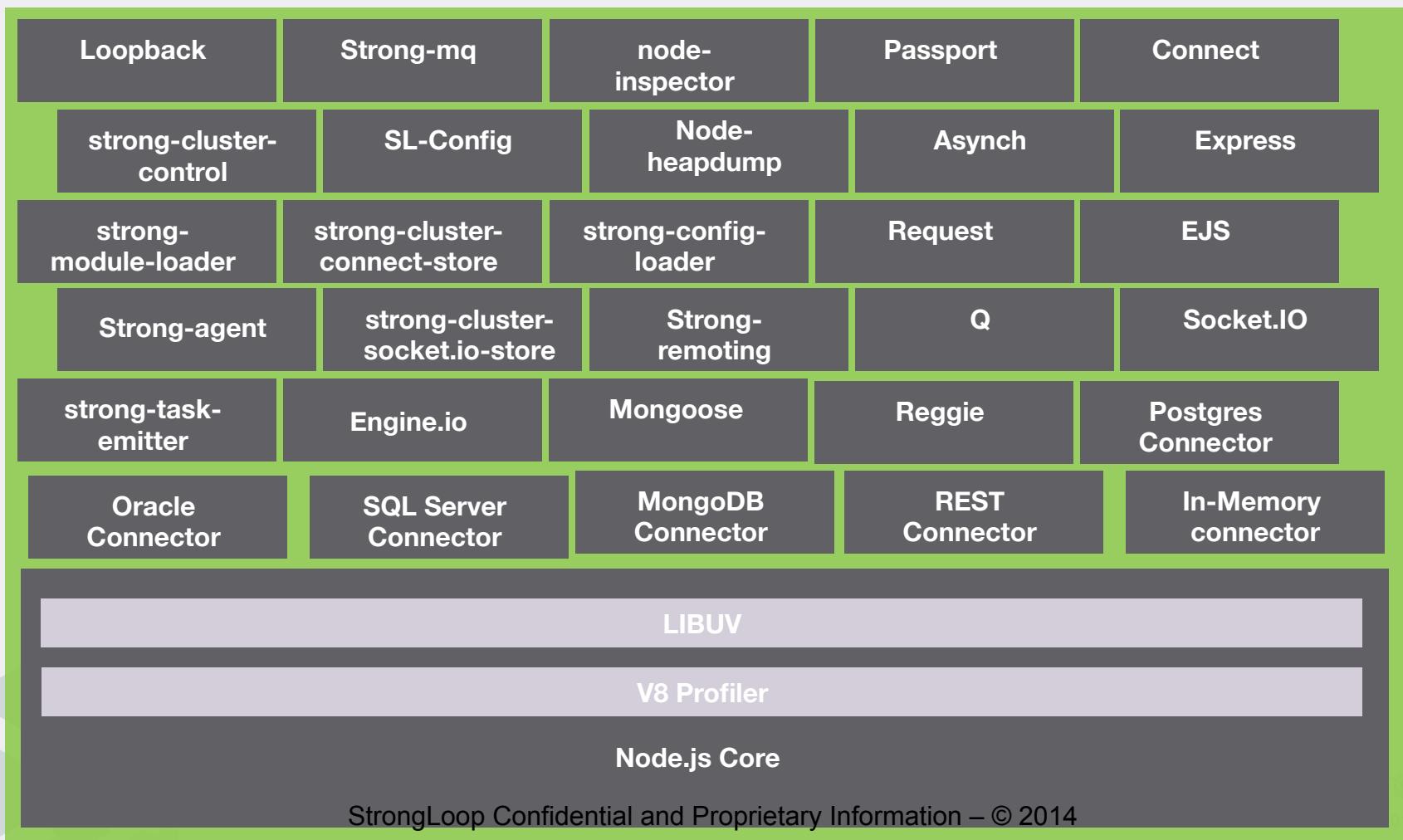
StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop 
© 2013

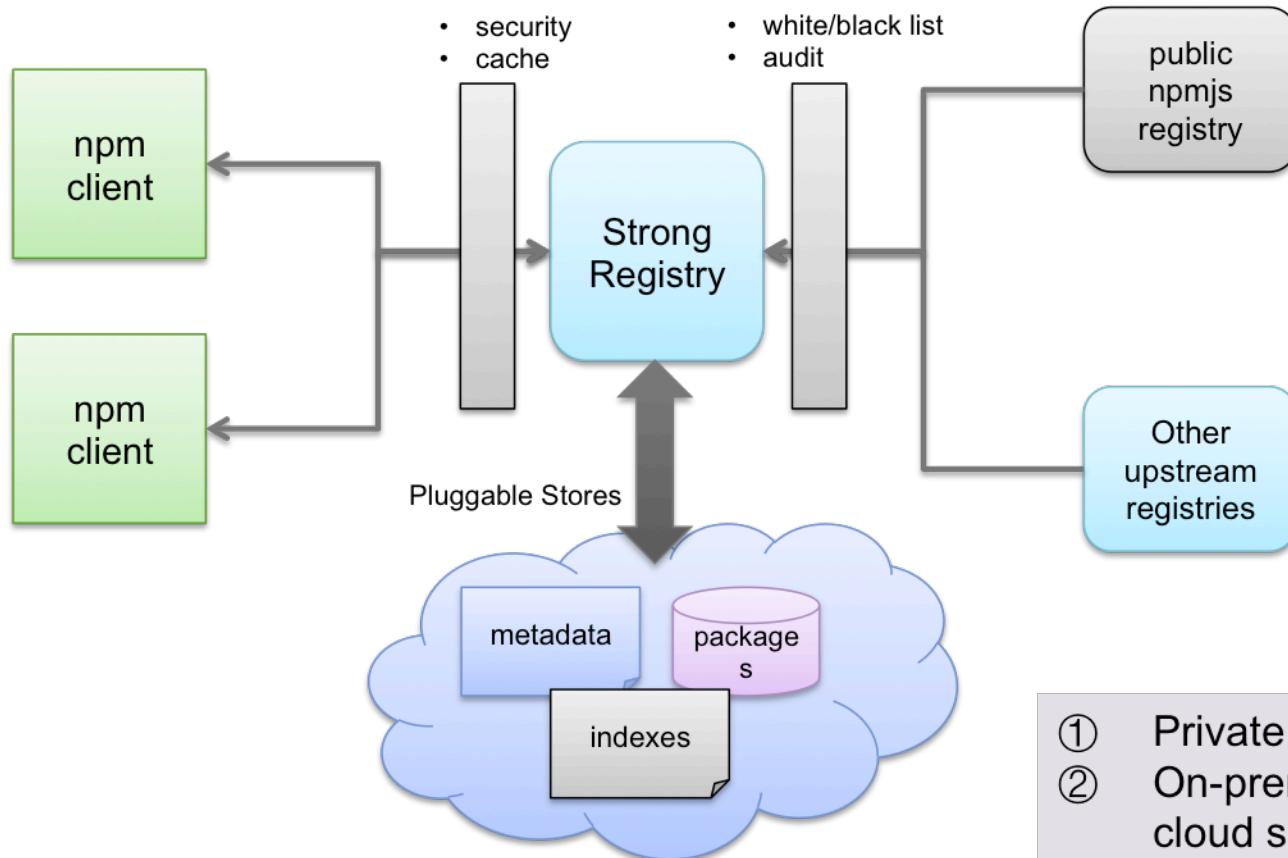
Platform support & DevOps Tool-chains

Node.js support (Sample modules set)

- ① Multi-platform (Windows, Unix, Solaris, Mac) support
- ② On-premises, private or public cloud support
- ③ Certified Node.js curated modules and ecosystem
- ④ Commercial Enterprise supported, security updates



Private Registry

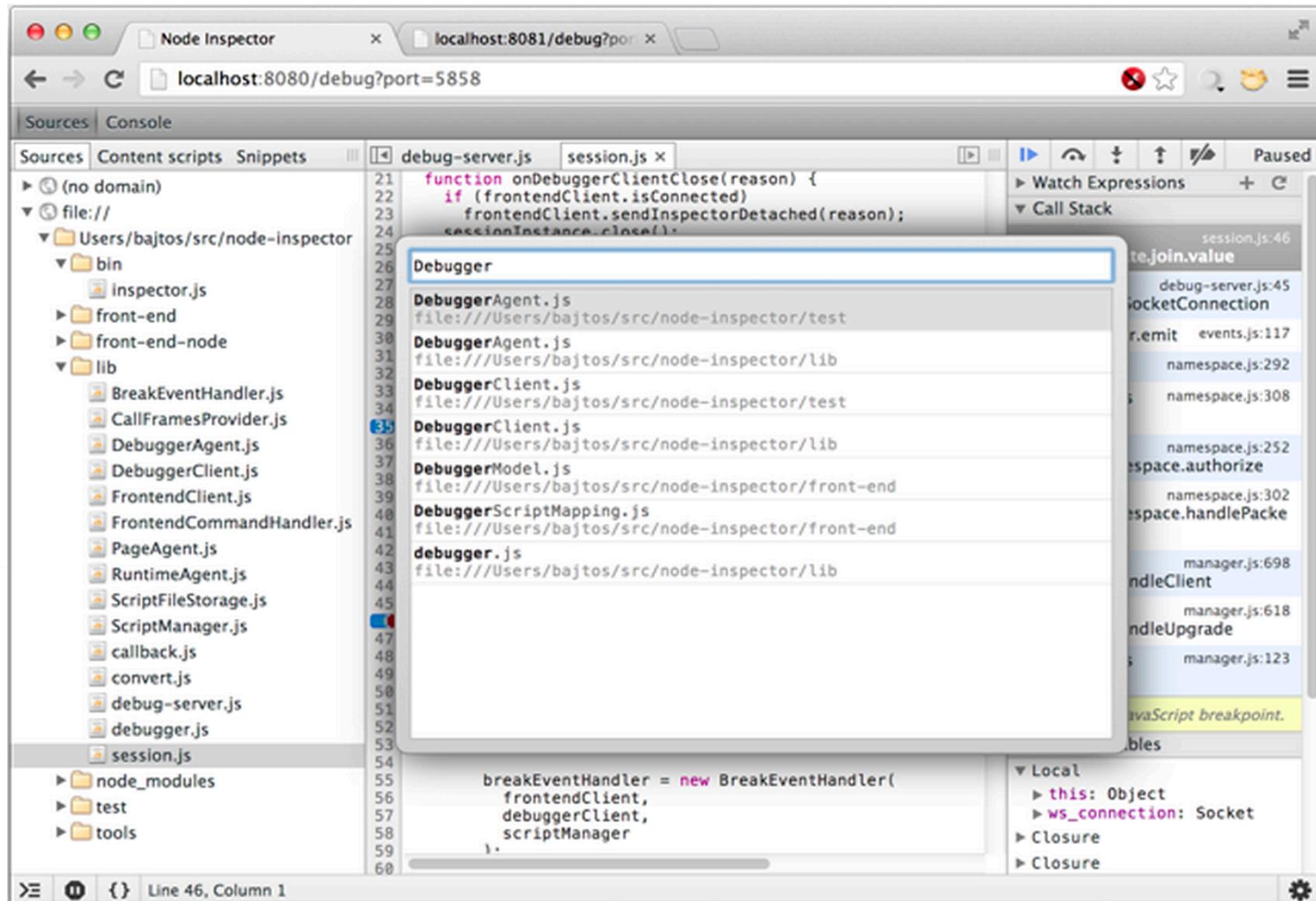


- ① Private & multi-registry
- ② On-premises, private or public cloud support
- ③ Module Whitelisting
- ④ Audit
- ⑤ Configuration Mgmt.
- ⑥ CI and Build Mgmt.



Debugging

Marquee – Chrome Dev Tools, Remote



Debugging server side clustered apps

Debugging single process

```
slc debug app.js
```

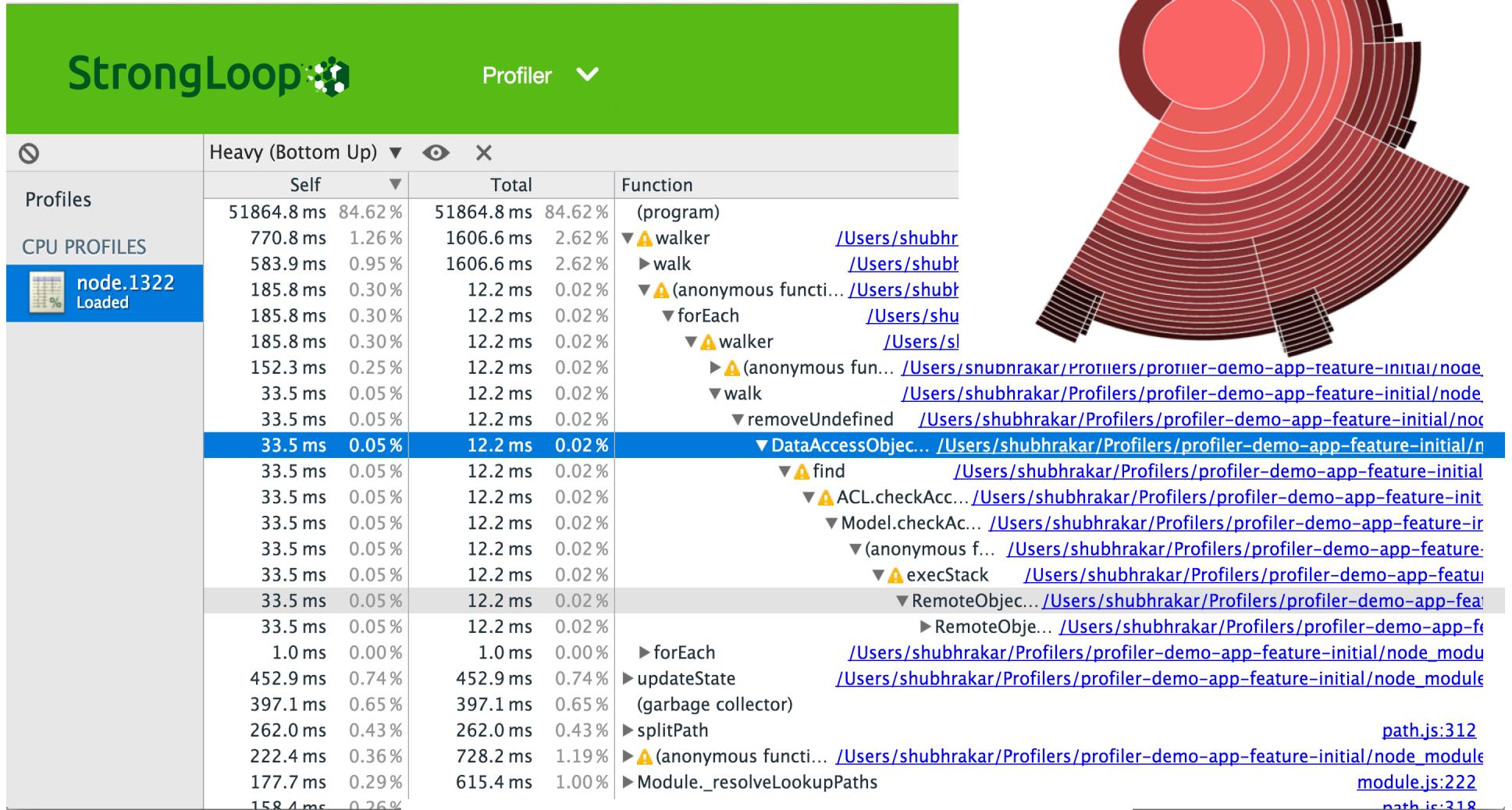
Debugging clustered process in V11

```
slc debug app.js  
http://localhost:8080/debug?port=<5858+ID-of-process>
```

Debugging clustered process in V10

```
%node debug -p <PID>  
connecting...ok  
...  
debug>quit  
%node-inspector  
http://127.0.0.1:8080/debug?port =5858  
Or  
process._debugPort = 5858 + cluster.worker.id
```

Profilers



Does Node have memory leaks ?

StrongLoop Profiler

Profiles

HEAP SNAPSHTOS

heapdump-1 2.3 MB

Constructor	Distance	Objects.	Shallow Size	Retained Size
ChocolateClass	8	501	37%	204 024
ChocolateClass @9411	8		24	0%
__proto__ :: ChocolateClass @9413	8		24	0%
constructor :: function ChocolateClass() @17879	7		72	0%
ChocolateClass :: function ChocolateClass() @17879	7		72	0%
prototype :: ChocolateClass @9413	8		24	0%
►shared :: ChocolateClass @11813	8		160	0%
►sugar :: Array @17885	7		32	0%
elements :: (object elements)[] @35383	8		80 160	3%
▼[0] :: ChocolateClass @19137	8		24	0%

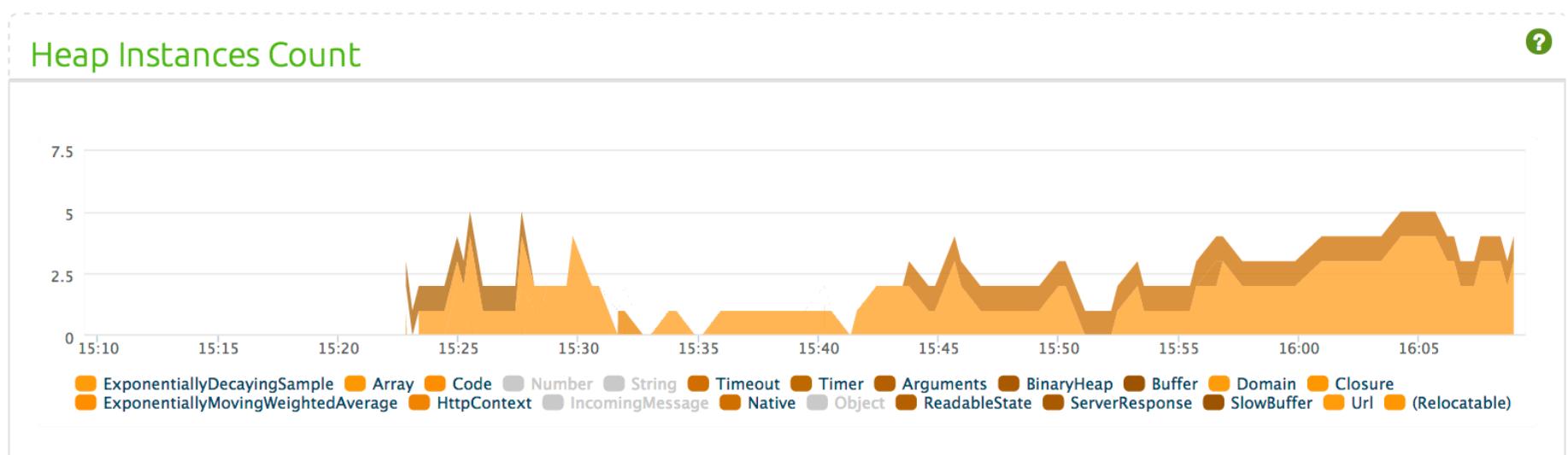
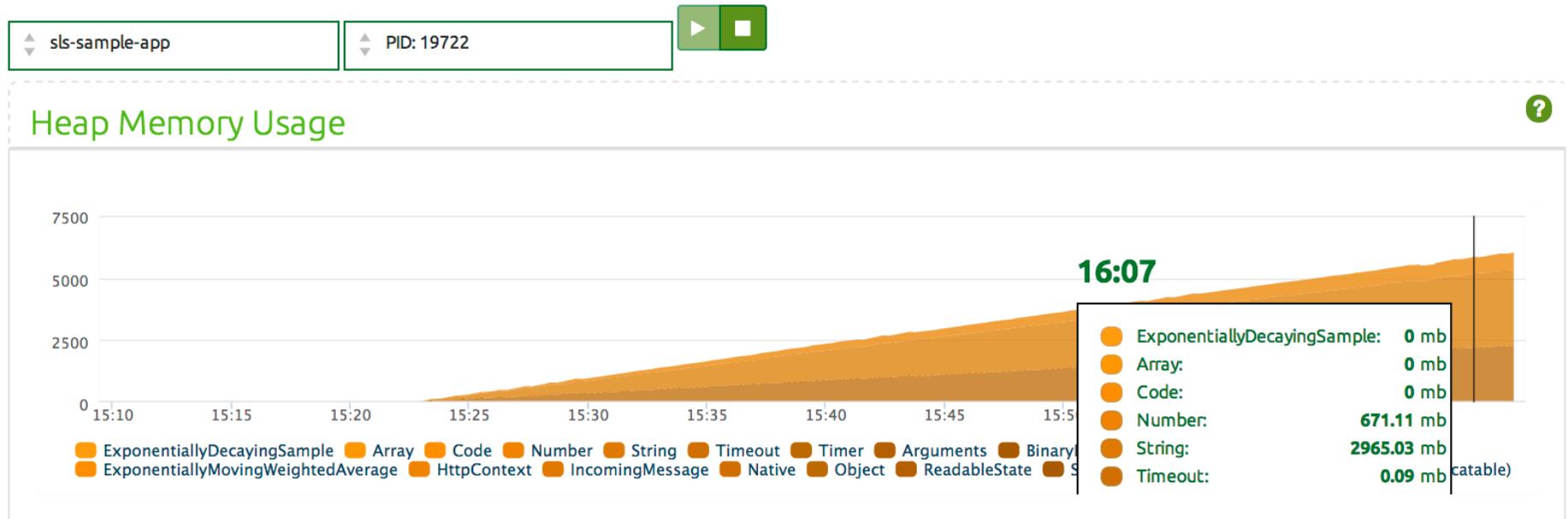
Object	Distan.▲	Shallow Size	Retained Size
►prototype in function ChocolateClass() @17879	7	72	0%
ChocolateClass in function() @38509	6	72	0%
callback in function wrapper() @38513	5	72	0%
►_onTimeout in Timeout @35453	4	136	0%
►_idlePrev in Timer @35961	3	32	0%
▼[56] in (Global handles) @27	2	0	0%
[9] in (GC roots) @3	1	0	532 640 22%
_idleNext in Timeout @35453	4	136	0%
_idlePrev in Timeout @35453	4	136	0%
►[1000] in @12337	5	24	0%
►idleNext in Timer @35961	3	32	0%

StrongLoop OSS - HeapSnapshots

slc runctl heap-snapshot ID

Sudden leaks

Heap Profiler [BETA]



heapdump for V8 snapshots

- ◆ npm install heapdump
- ◆ Add to app : var heapdump = require('heapdump')
- ◆ Method 1 : writeSnapshot

```
var heapdump = require('heapdump')
...
heapdump.writeSnapshot()
```

- ◆ Method 2 : SIGUSR2 (Unix only)

```
kill -USR2 <pid>
```

- ◆ Make sure your directory is writable

```
process.chdir('/path/to/writeable/dir')
```



Heap Snapshot strategies

- ◆ Programmatic heap snapshots (timer based)

```
var heapdump = require('heapdump')
...
setInterval(function () {
  heapdump.writeSnapshot()
}, 6000 * 30) <strong>(1)</strong>
```

- ◆ Programmatic heap snapshots (threshold based)

```
var heapdump = require('heapdump')
var nextMBThreshold = 0 <strong>(1)</strong>

setInterval(function () {
  var memMB = process.memoryUsage().rss / 1048576 <strong>(2)</strong>
  if (memMB > nextMBThreshold) { <strong>(3)</strong>
    heapdump.writeSnapshot()
    nextMBThreshold += 100
  }
}, 6000 * 2) <strong>(4)</strong>
```

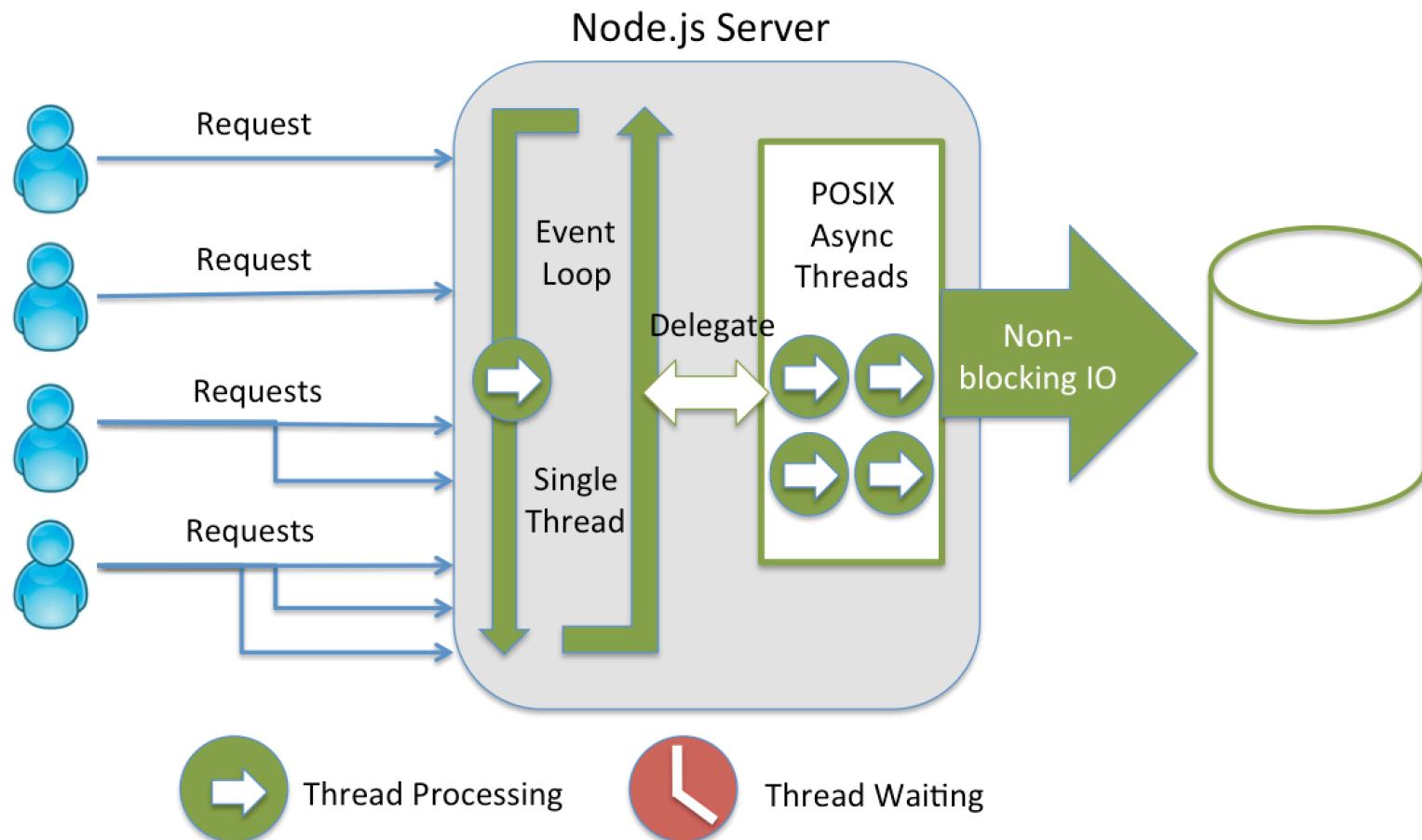


How we fixed a real production problem



StrongLoop™

Don't Block the EventLoop



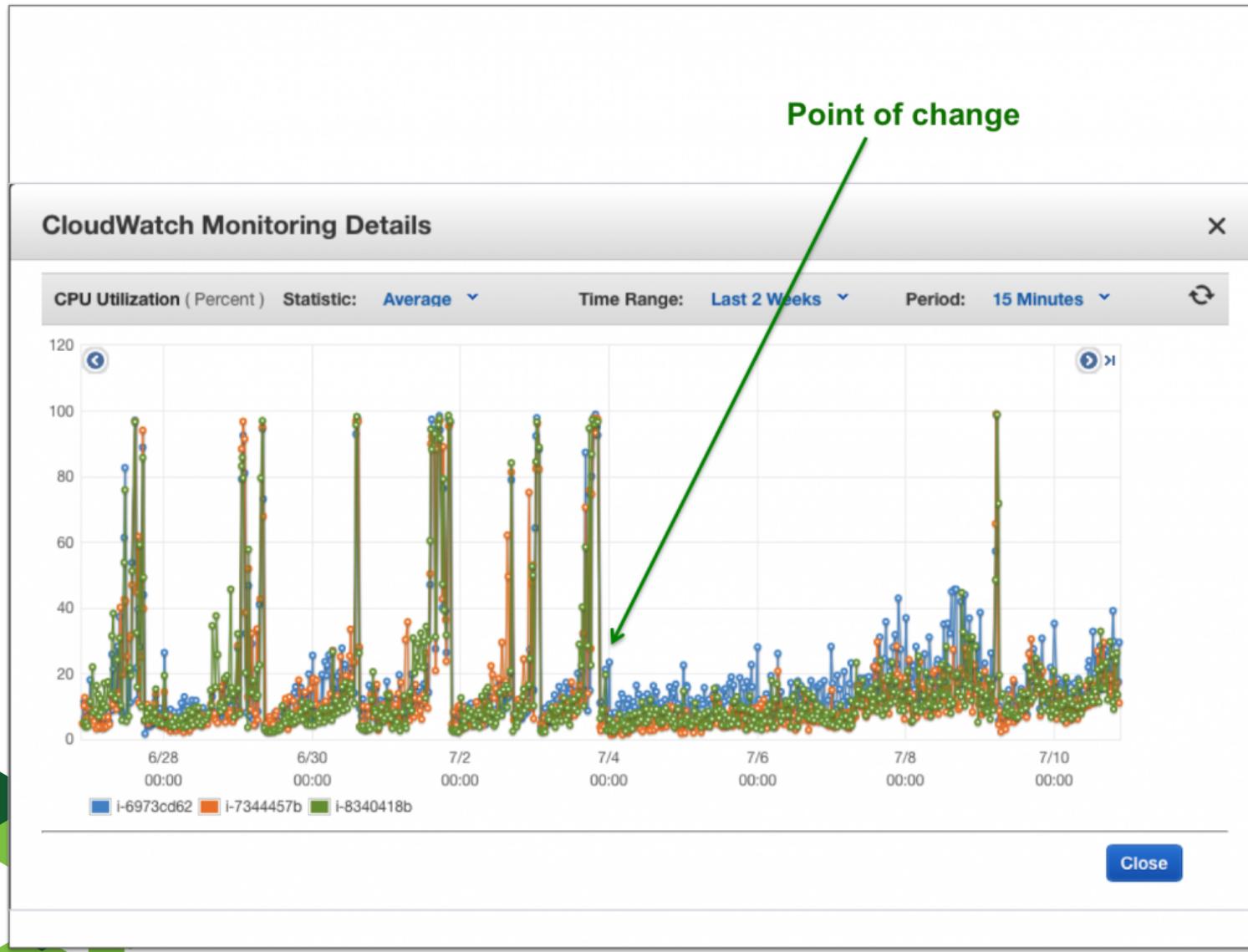
Blocked event loop in Meteor atmosphere

```
ticks parent name
2274 7.3% v8::internal::Isolate::FindOrAllocatePerThreadDataForThisThread()
1325 58.3% LazyCompile: ~<anonymous> packages/meteor.js:683
1325 100.0% LazyCompile: _tickCallback node.js:399
```

- node-fibers implements co-routines. Meteor uses this to hack local thread storage allowing V8 to run multiple execution contexts each mapped to a co-routine.
- FindOrAllocatePerThreadDataForThisThread()** used in switching context between co-routines
- Co-routines are cooperative; the current coroutine has to yield control before another one can run and that is what Meteor does in its process.nextTick() callback; it essentially builds concurrent (but not parallel) green threads on a round-robin scheduler
- Too many tiny tasks and not one long running one** was blocking the event loop
- process.nextTick() has a failsafe mechanism where it will process “x” tick callbacks before deferring the remaining ones to the next event loop tick.
- Native MongoDB driver disabled the failsafe to silence a warning message in node v0.10 about maxtickDepth being reached

The solution

- ◆ The workaround: switch from process.nextTick() to setImmediate()



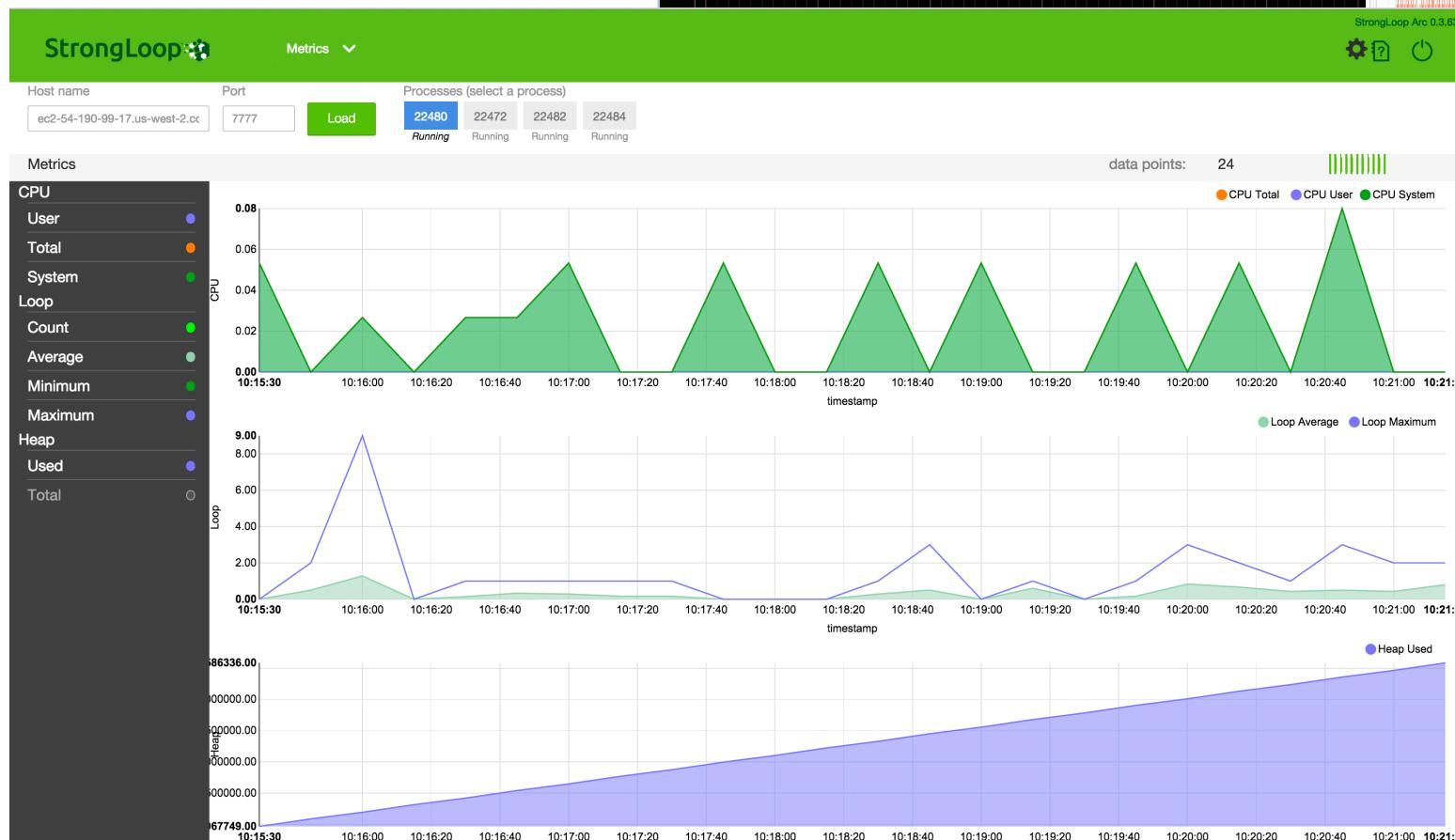
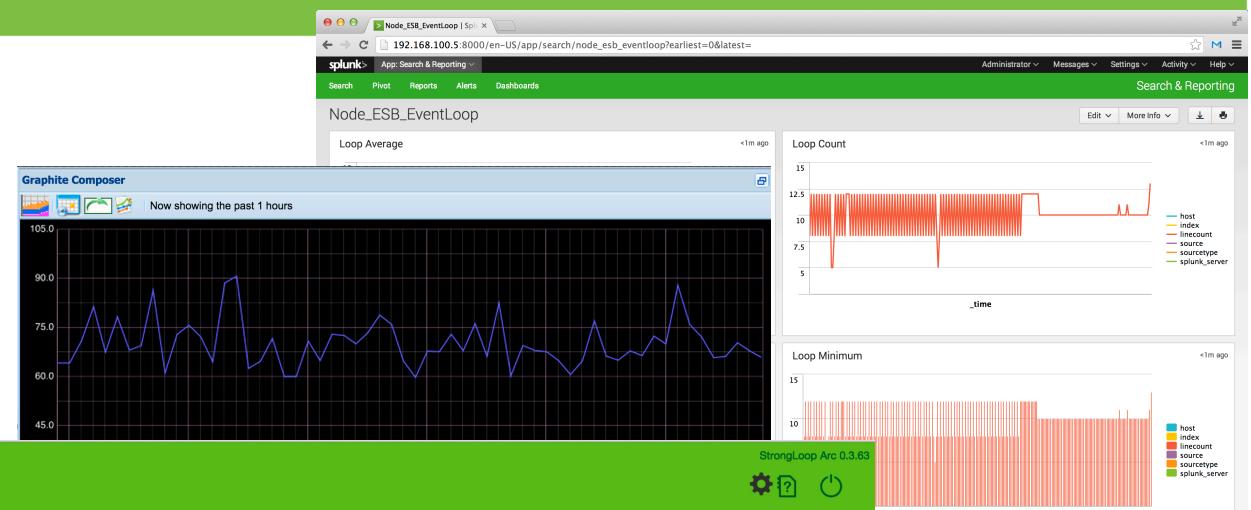
Performance Monitoring



StrongLoop™

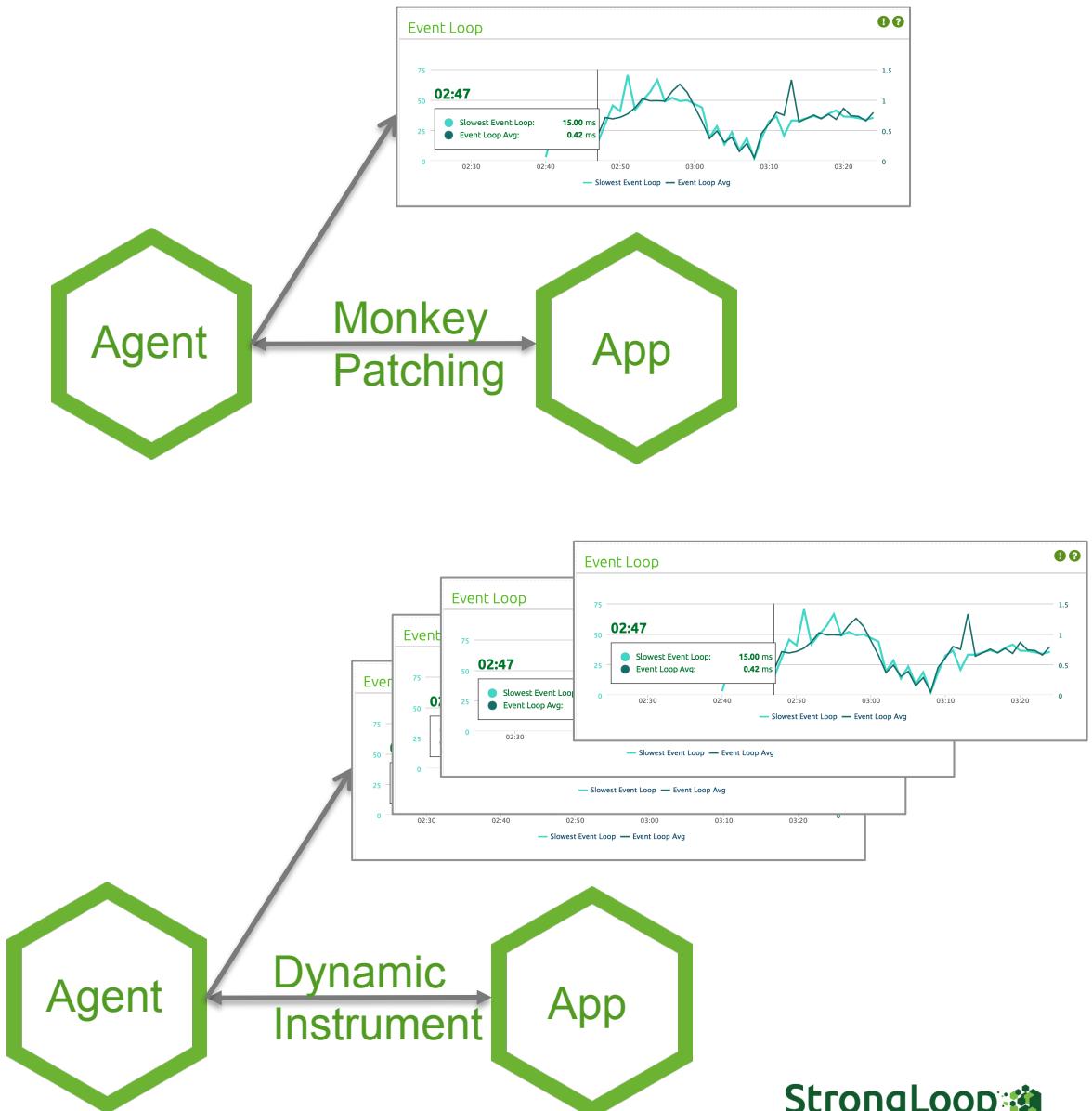
Production Monitoring & 3rd Party integration

- Local / Hosted
- Graphite
- Splunk
- Datadog
- CA Introscope



Dynamic Instrumentation

- Live Edit
- Line Level Instrumentation
- Any package, any framework, any code
- Custom logic
- Counters, Gauges and Timers
- HA rollback



Can Errors be stitched into a trace and recovered ?

Enter StrongLoop / Zones



Long error traces

Zones are enabled. See <http://strongloop.com/zone> for more information.

ReferenceError: function_that_doesnt_exist is not defined

```
at null.<anonymous> (/Users/shubhrakar/zone/showcase/long-stack/long-stack.js:20:5)
at Zone._apply (/Users/shubhrakar/zone/lib/zone.js:596:15)
at Zone.apply (/Users/shubhrakar/zone/lib/zone.js:620:23)
at ZoneCallback.call [as apply] (/Users/shubhrakar/zone/lib/zone-callback.js:94:31)
at wrapper (/Users/shubhrakar/zone/lib/zone.js:758:21)
at Timer.listOnTimeout (timers.js:133:15)
```

In zone: AsyncFailZone

```
at defineZone (/Users/shubhrakar/zone/lib/zone.js:321:13)
at Zone.<anonymous> (/Users/shubhrakar/zone/showcase/long-stack/long-stack.js:14:5)
at Zone._apply (/Users/shubhrakar/zone/lib/zone.js:596:15)
at Zone.apply (/Users/shubhrakar/zone/lib/zone.js:620:23)
at Zone.run [as _run] (/Users/shubhrakar/zone/lib/zone.js:692:21)
at new Zone (/Users/shubhrakar/zone/lib/zone.js:238:10)
at Zone.create (/Users/shubhrakar/zone/lib/zone.js:274:10)
at Zone.createMiddleZone (/Users/shubhrakar/zone/showcase/long-stack/long-stack.js:12:8)
at Zone._apply (/Users/shubhrakar/zone/lib/zone.js:596:15)
at Zone.apply (/Users/shubhrakar/zone/lib/zone.js:620:23)
```

In zone: Anonymous

```
at Zone.create (/Users/shubhrakar/zone/lib/zone.js:274:10)
at Zone.createMiddleZone (/Users/shubhrakar/zone/showcase/long-stack/long-stack.js:12:8)
at Zone._apply (/Users/shubhrakar/zone/lib/zone.js:596:15)
at Zone.apply (/Users/shubhrakar/zone/lib/zone.js:620:23)
at processCallbacks (/Users/shubhrakar/zone/lib/scheduler.js:47:10)
at processQueues (/Users/shubhrakar/zone/lib/scheduler.js:67:5)
at EventEmitter._tickCallback (node.js:372:11)
at Function.Module.runMain (module.js:503:11)
at startup (node.js:124:16)
at node.js:842:3
```

In zone: Outer

```
at Zone.create (/Users/shubhrakar/zone/lib/zone.js:274:10)
at Object.<anonymous> (/Users/shubhrakar/zone/showcase/long-stack/long-stack.js:7:6)
at Module._compile (module.js:460:26)
at Object.Module._extensions..js (module.js:478:10)
```



© 2013

Runtime Mgmt. & Scaling

Build & Deploy

StrongLoop  Build & Deploy  StrongLoop Studio Arc 0.3.0  

Git Universal ?

Build git ?

Git current branch:
Your current directory

Git deploy branch:
deploy

Build

Build Source ?

New  Existing

Deploy git ?

Git deploy branch:
deploy

Hostname: host1.example.com Port: 8000 Processes: 4

Deploy



StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop 
© 2013

Cluster and runtime scaling

Marquee – HA, Auto-scale, Hot Deploy

Cluster Control

Select Application:

- cluster-test0
- Master PID: 11249

Scale to Size:

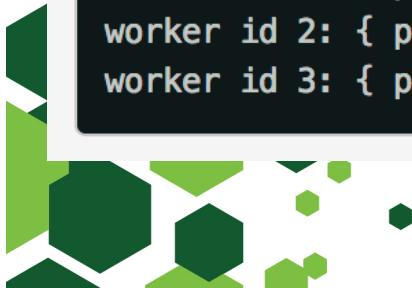
4



Control Workers:

filter by PID		<input type="button" value="Restart All"/>
Worker 1	PID: 11255	<input type="button" value="x"/> <input type="button" value="▼"/>
Worker 2	PID: 11263	<input type="button" value="x"/> <input type="button" value="▼"/>
Worker 5	PID: 11988	<input type="button" value="x"/> <input type="button" value="▼"/>
Worker 6	PID: 11990	<input type="button" value="x"/> <input type="button" value="▼"/>

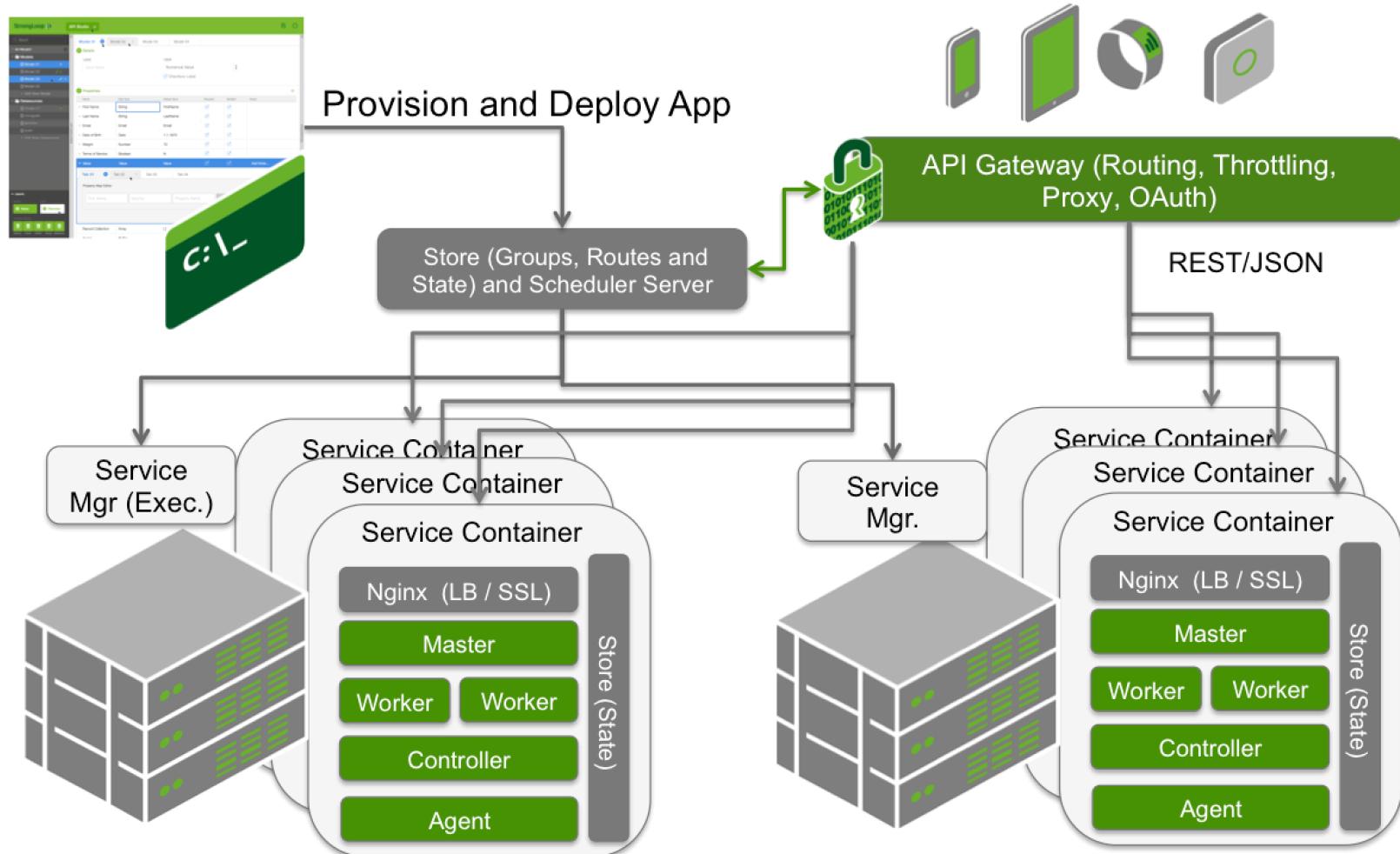
```
$ slc clusterctl set-size 4
$ slc clusterctl status
worker count: 4
worker id 0: { pid: 7696 }
worker id 1: { pid: 7703 }
worker id 2: { pid: 7705 }
worker id 3: { pid: 7707 }
```



StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop 
© 2013

MESH – Micro-services Deploy on Scale



StrongLoop Confidential and Proprietary Information – © 2014

StrongLoop 
© 2013



Thank you!