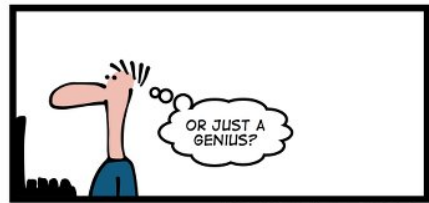
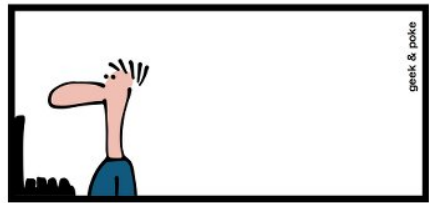


# Battle Ready! - Hardening Node.js for the Enterprise

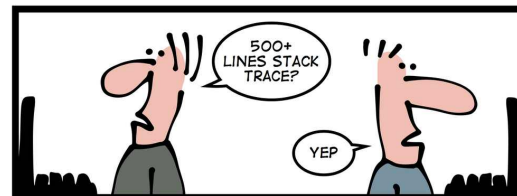
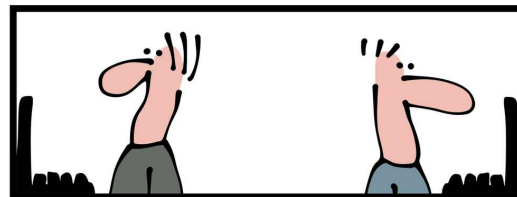
**Jeff Barczewski**

[codewinds.com](http://codewinds.com)

*CodeWinds*



YESTERDAYS REGEX



"LEAKY ABSTRACTIONS"

See "The Law of Leaky Abstractions" from Joel Spolsky  
(<http://www.joelonsoftware.com/articles/LeakyAbstractions.html>)  
Over 10 years old but still true.

# Why build scalable, durable apps?



**Jeff Barczewski**

@jeffbski

It is interesting to see who has scalable web architectures and who doesn't. Kmart's site melted down for me today. Try [#nodejs](#)



RETWEETS

2

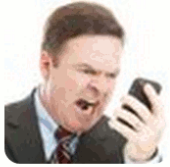
FAVORITES

2



2:38 PM - 29 Nov 2013





**PHP CEO**

@PHP\_CEO



Follow

HIRING PRO TIP: HIPPIES ARE NOT THE  
SAME AS HIPSTERS

I HAVE MADE THIS MISTAKE AND WE NOW  
HAVE A FRONTEND MVC FRAMEWORK  
WRITTEN IN FORTRAN



RETWEETS

1,463

FAVORITES

825



# Who am I?

- 25 yrs professional developer
- 3 yrs full time with Node.js
- MasterCard, Elsevier, RGA, USAF
- CodeWinds Podcast and Training



# Elsevier Education Platform

- Rapid Development Apps
- Node.js
- Isomorphic Javascript

# MasterCard CaaS

- Crypto as a Service
- Node.js plus add-on HSM
- REST API
- Highly scalable and available
- Reporting, Administration

# MasterCard / Apple Pay Launch

- Complex deployment, many third parties
- Field testing
- US public launch - "It just worked"

# Other Enterprise Use

- Walmart
- Pay Pal, Linkedin
- Yahoo, ebay, Uber, Netflix
- <http://nodejs.org/industry/>

# Legacy Code & Stagnation

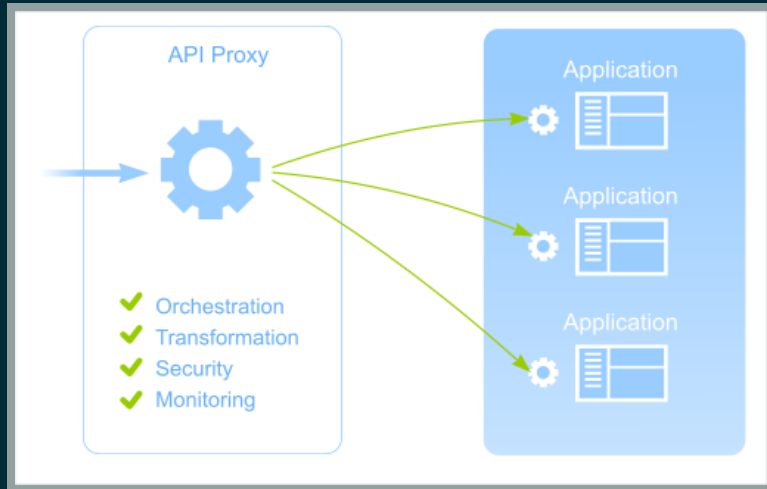
"Inspire with your actions"

# How to Introduce Node.js

## Simple beginnings

- CLI tools
- Batch processing
- Admin applications
- Green field projects, REST, SPA

# Smart API reverse proxy



- SSO, batching
- Orchestration, caching
- Device specific content
- Migration, AB Testing

API Proxy by Yury Baranovsky



**Inspired**

# F15 Eagle



F-15E Strike Eagle by Gerry Metzler - IMG\_214 Licensed CC BY-SA 2.0



Afghanistan, F-15E 391st" by Staff Sgt. Aaron Allmon (USAF) - Src. Public domain

# F15 Video

- Full F15 video from 67FS - I shared an edited portion of this video in my talk

**Life happens**

# F15 Single Wing Landing



# Traits that inspire me

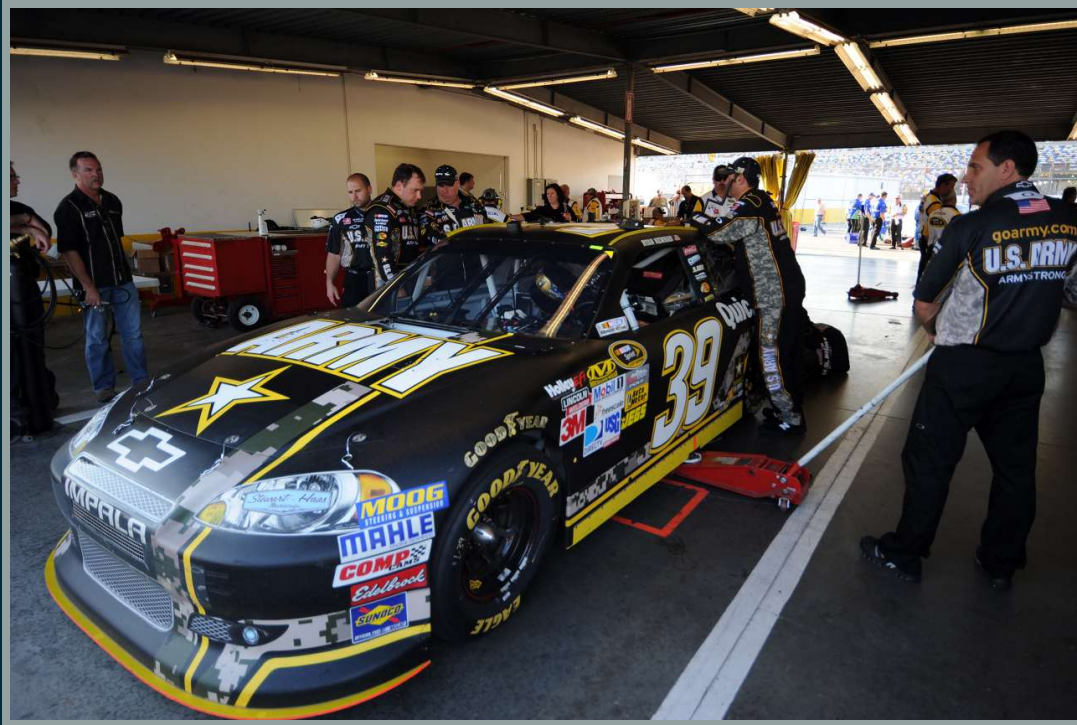
- Performant
- Adaptability
- Durability
- Trusted

# Design and Foundations

## What frameworks and tools?

- Hapi, Express, Restify
- Is it battle tested? Quality? Community? License?
- Prototype with API
- Joi, Wreck

# Express & Hapi



Army Car Garage by U.S. Army Licensed CC BY-SA 2.0



Stingray by Jeff Barczewski



# Express Example p1

```
var express = require('express');  
var compression = require('compression')  
var logger = require('morgan');  
var cookieParser = require('cookie-parser');  
var bodyParser = require('body-parser');  
var app = express();  
app.use(compression())  
app.use(logger('dev'));  
app.use(bodyParser.json());  
app.use(bodyParser.urlencoded({ extended: false }));  
app.use(cookieParser());
```

# Express Example p2

```
app.get('/', function (req, res) {
  res.send({ hello: 'world' });
});

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  var err = new Error('Not Found');
  err.status = 404;
  next(err);
});

app.listen(3000, function(){
  console.log('Express server listening on port ' + 3000); });
```

# Hapi Example

```
var Hapi = require('hapi');  
var server = new Hapi.Server(3000);  
server.route({  
  method: 'GET', path: '/',  
  handler: function (request, reply) {  
    reply({ hello: 'world' });  
  }  
});  
server.start(function () {  
  console.log('Server running at:', server.info.uri);  
});
```

# Design and Foundations

## Where to store data?

- Redis - caching, sessions, queues
- REST services - SQL (postgres), noSQL (riak, dynamo, leveldb)
- async libraries, uses, verify
- Sharding plan

# Design and Foundations

## Reducing risk

- RDD, Test API, CI
- Git pull requests, shrinkwrap
- jshint, markdown, jsdoc
- Node.js / JS training

# How do we scale up?

- Node.js evented architecture
- Cluster, Child processes, Containers, Zones
- Load balancer, nginx, HA proxy, apache

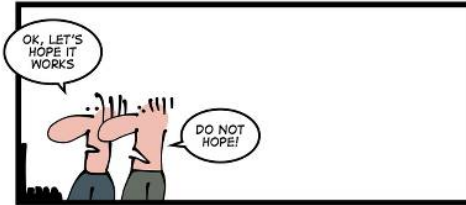
# How do we know it will scale?

- Load Test, stress test, vary workload
- bench-rest, wrk, ab
- load runner, blitz.io
- benchmark frequently

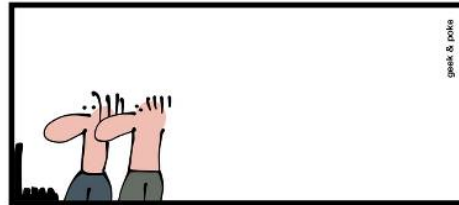
# bench-rest

```
// unique PUT and then a GET for each iteration  
var flow = {  
  main: [  
    { put: 'http://localhost:8000/foo_#{INDEX}',  
      json: 'mydata_#{INDEX}' },  
    { get: 'http://localhost:8000/foo_#{INDEX}' }  
  ]  
};  
module.exports = flow;
```





NEVER RELY JUST ON HOPE



ALWAYS DO MORE!

# Resilience

## How do we make this durable?

- child processes, cluster\_master\_ext
- Detect process exit, restart
- DB failures, reconnect generic-pool, backoff
- Monit, upstart, forever, nscale

# cluster\_master\_ext

```
var clusterMaster = require("cluster-master-ext");  
  
clusterMaster({  
  exec: "worker.js", // script to run  
  size: 5, // number of workers  
  env: { SOME: "environment_vars" },  
  args: [ "--deep", "doop" ],  
  silent: true,  
  signals: true  
});
```

# How to make this hot deployable?

- Rolling restart
- Handle shutdown, exit cleanly, fail safe timeout
- Signals: TERM, INT - shutdown, HUP - rolling restart

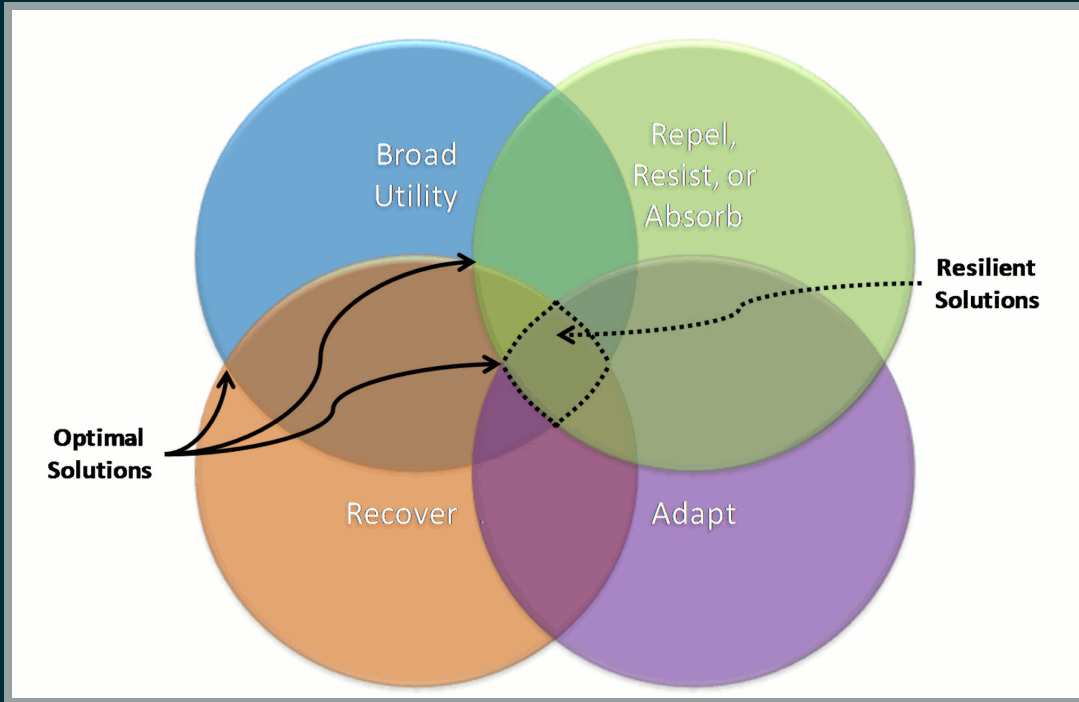
# How do we monitor?

- Logging? Events?
- Bunyan, jeffbski/hapi-bunyan-lite
- Monit, upstart, forever, nscale
- Notifying, dtrace, vasync
- Charting - Rickshaw, zag, statsd/graphite, influxdb

# Demonstration

**True Friend**

# Resilient Solutions



- Strong
- Reliable
- Trusted
- Adaptable



# To learn more

- <http://codewinds.com/nodevember>
- CodeWinds - Resilient Systems
- [jeff@codewinds.com](mailto:jeff@codewinds.com)
- [@codewinds](#) [@jeffbski](#)