

CONSUMING REST

APIS

FOR ALL INTERPRETATIONS OF REST

@darrel_miller

http://

203

Non Authoritative

TODAY'S MENU

Appetizers

API Economy

Client SDKs

Main Course

Interpretations of REST

Change is inevitable and good

Code reuse on the web

The secret of REST clients

Dessert

Samples and Libraries to explore



API ECONOMY



HIDING THE GOODNESS



CONTROL IS CRITICAL



QUALITY + QUANTITY = \$\$\$



CHEF'S SUGGESTION



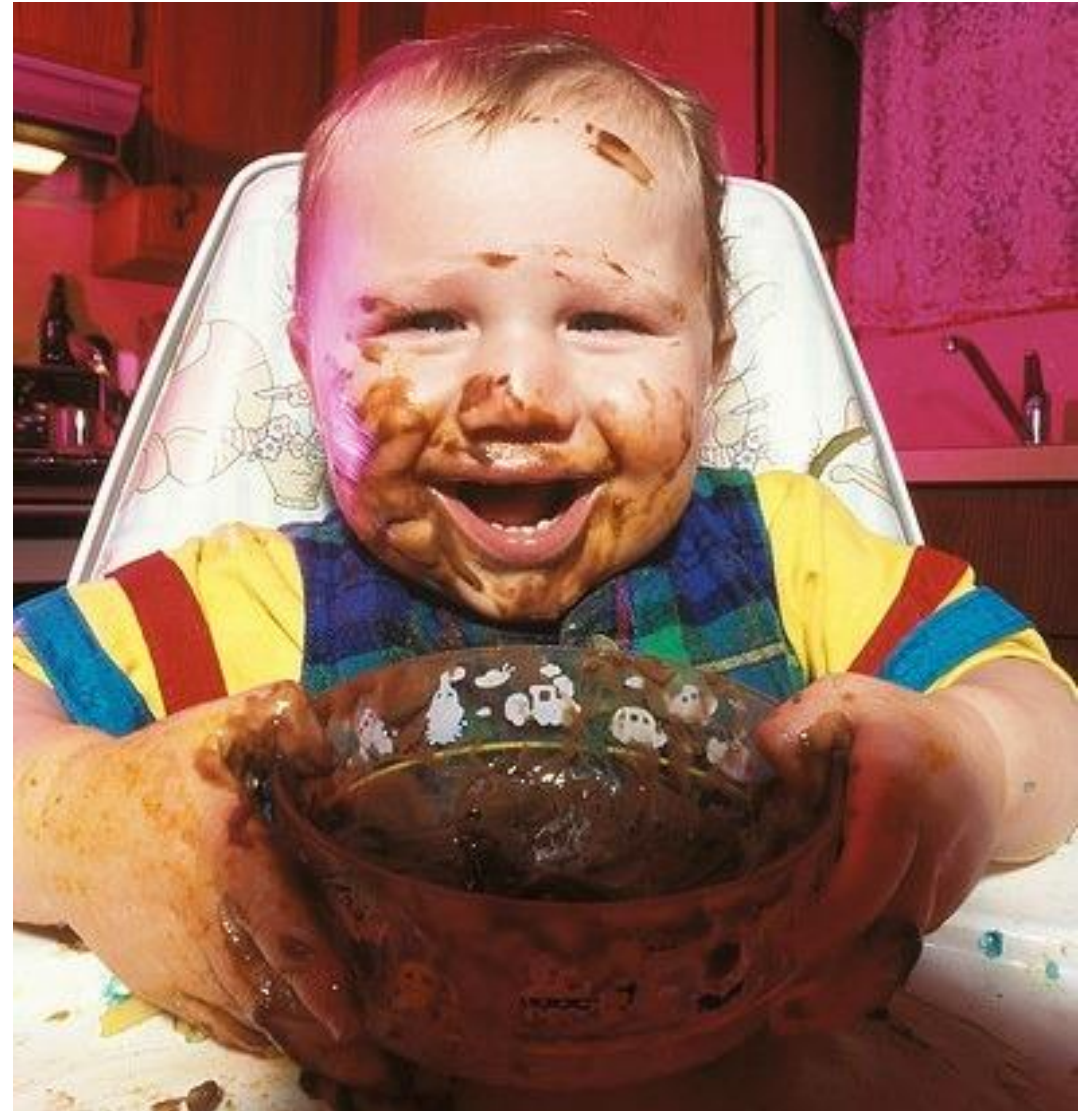
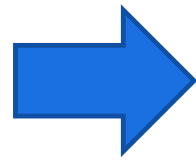
DON'T ASSUME THE PROVIDED SDK IS THE BEST OPTION

UNDERSTAND YOUR DEPLOYMENT CONSTRAINTS

WHAT DO YOU MEAN BY REST?

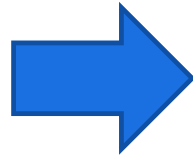


DESIRED EFFECT?



DESIRED EFFECT?

- Client/Server
- Stateless
- Caching
- Uniform Interface
- Layered
- Code On Demand



- Scalable
- Evolvable
- Fault Tolerant
- Composable

JSON-REST TASTES GREAT...

Poutine

Quick and Easy



Warning : Long term consumption may be detrimental to your health

ADD SOME HYPERMEDIA



APIS DONE OUR WAY



`application/vnd.github.v3+json`

`application/vnd.heroku+json; version=3`

GENERIC HYPERMEDIA

Mason

HAL

Hyper-Schema

Uber



Siren

JSON-LD

Json-api

CHEF'S SUGGESTION



BE AWARE OF THE EFFECTS OF YOUR DESIGN CHOICES

UNDERSTAND THE PURPOSE OF MEDIA TYPES

DATA ORIENTED APIS



SCENARIO BASED APIS



HYPERMEDIA DRIVEN RITUALS



CHEF'S SUGGESTION



DATA ORIENTED APIS ARE GREAT FOR QUICK, EASY AND INTERNAL
OR GOVERNMENT DATA

SCENARIO ORIENTED IF YOU WANT TO MAKE MONEY

CHANGE IS GOOD



START SMALL, EVOLVE QUICKLY



START SMALL, EVOLVE QUICKLY



START SMALL, EVOLVE QUICKLY



VERSIONING HURTS



A LITTLE PARANOIA CAN BE USEFUL



CHEF'S SUGGESTION



CHANGE IS GOOD

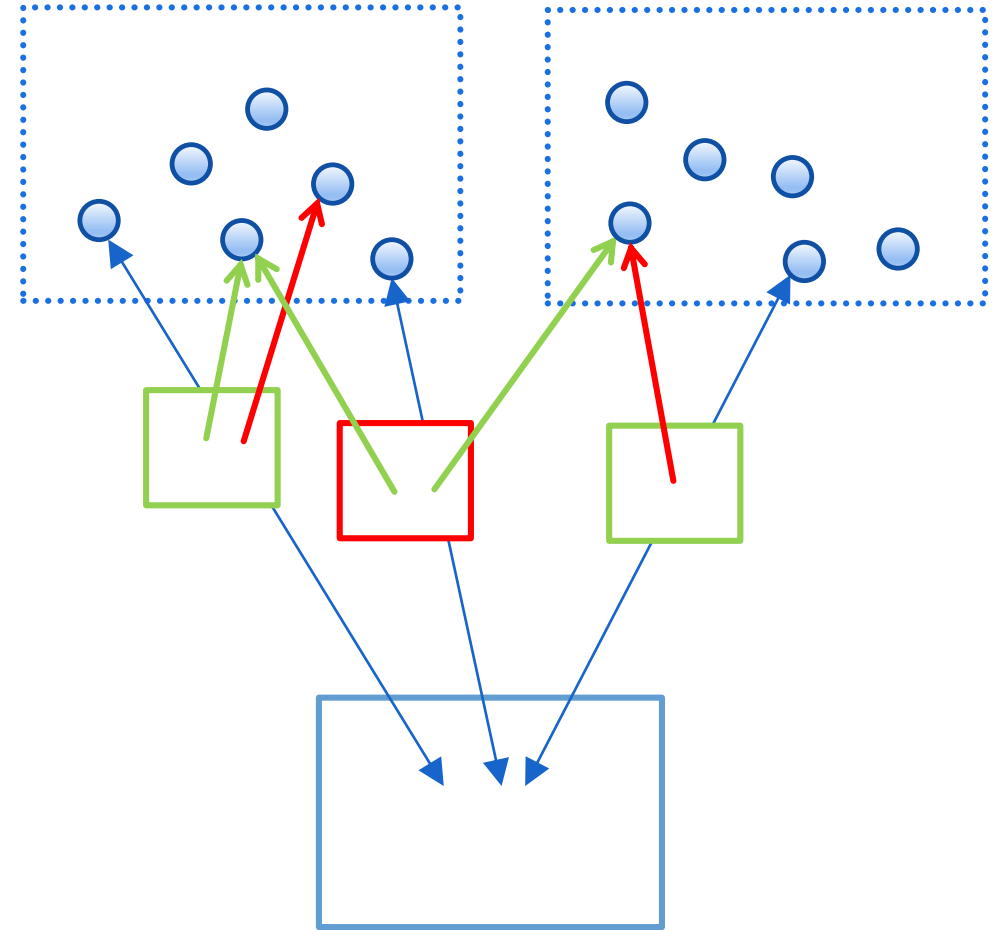
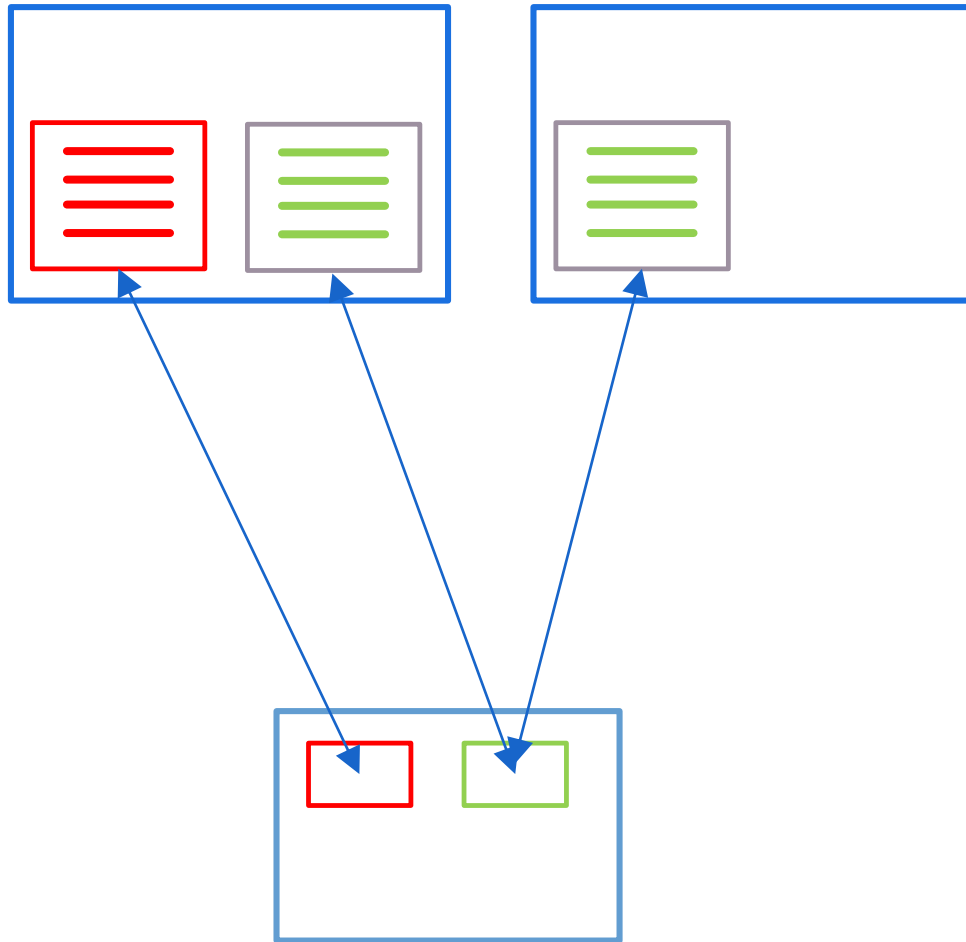
USE VERSIONS AS A LAST RESORT

SOFTWARE'S HOLY GRAIL

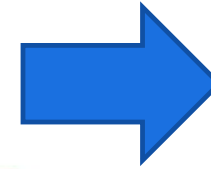
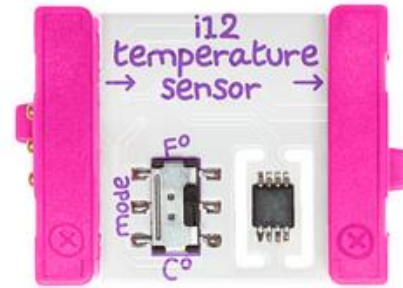
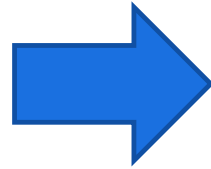
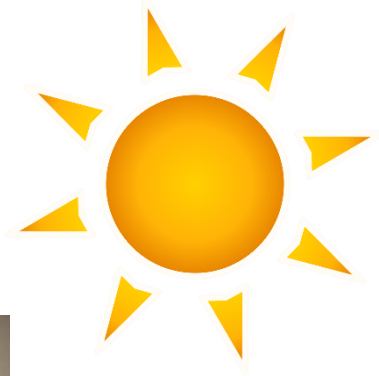


Achieving reuse on the web is different...

API REUSE VERSUS WEB REUSE



IGNORANCE IS BLISS



REUSABLE COMPONENTS



MAYBE ONE DAY...



BEWARE MEDIA TYPE EXPLOSION



SUPPORT MULTIPLE MEDIA TYPES



TYPED LINKS



CHEF'S SUGGESTION



CONSIDER HYPERMEDIA TO ENABLE CLIENT CODE REUSE

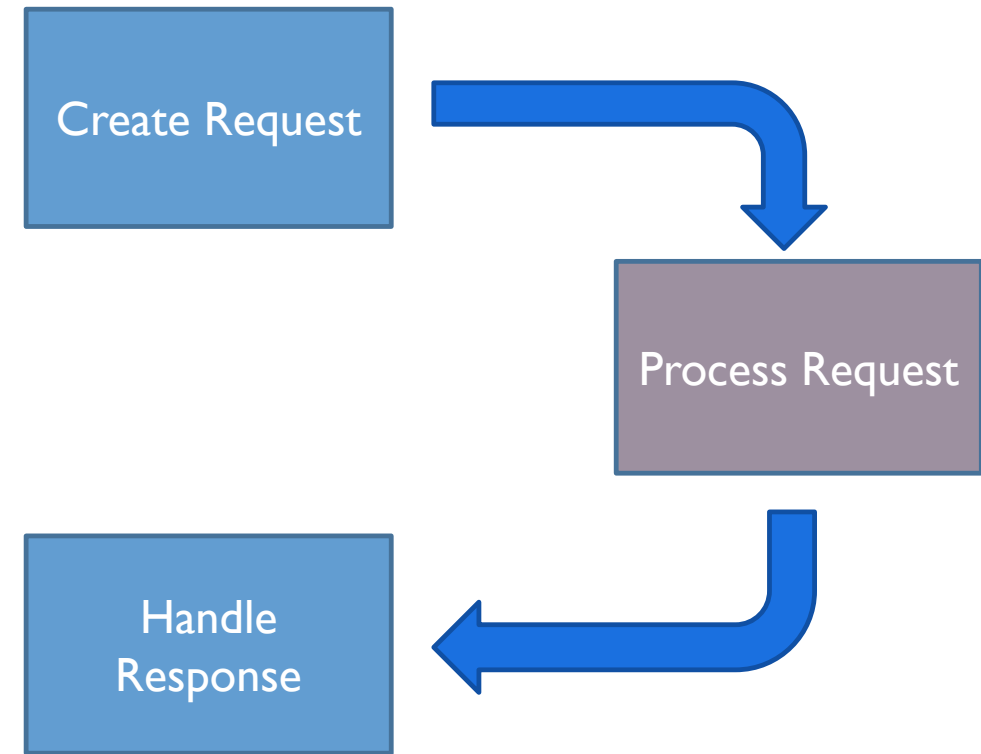
UNDERSTAND THE CONTRACTS YOU ARE USING

THE SECRET OF REST CLIENTS

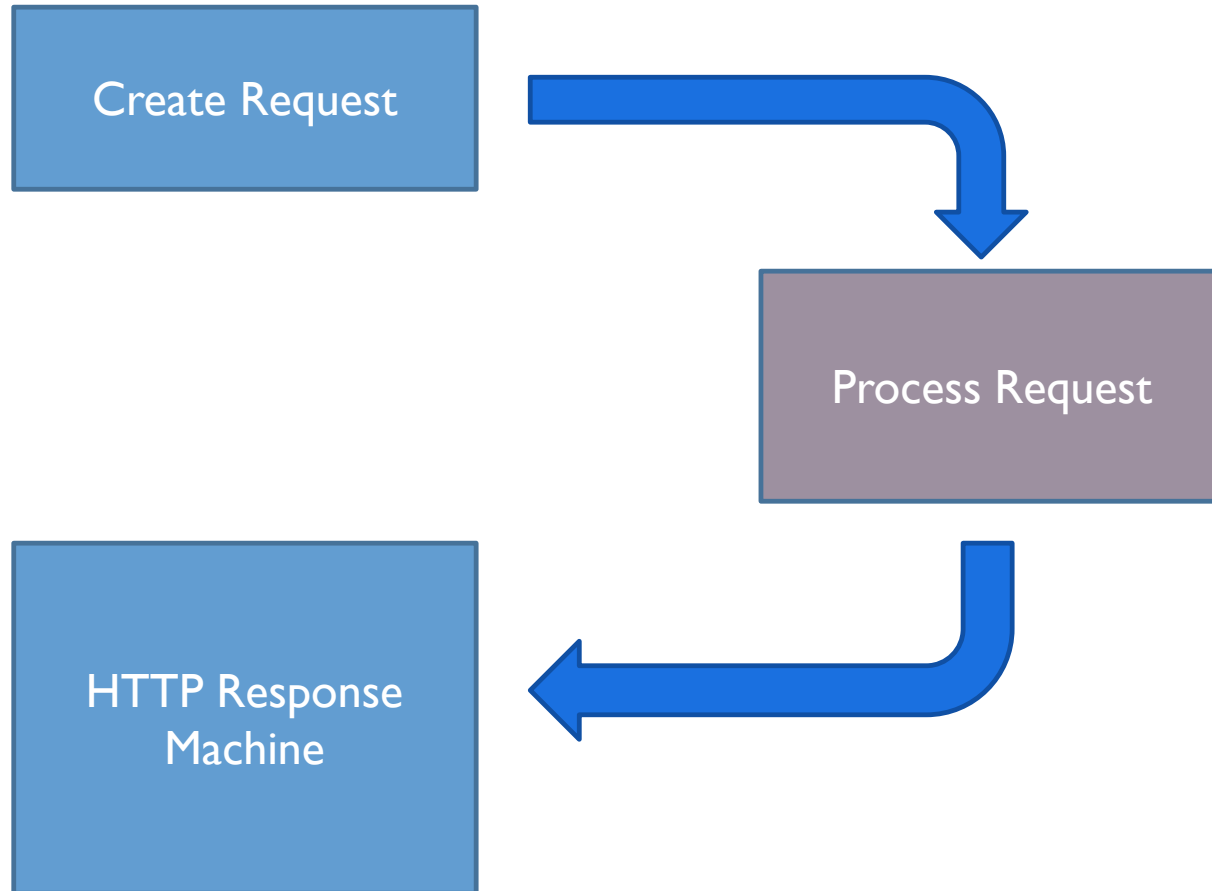


**Client code dictates
coupling**

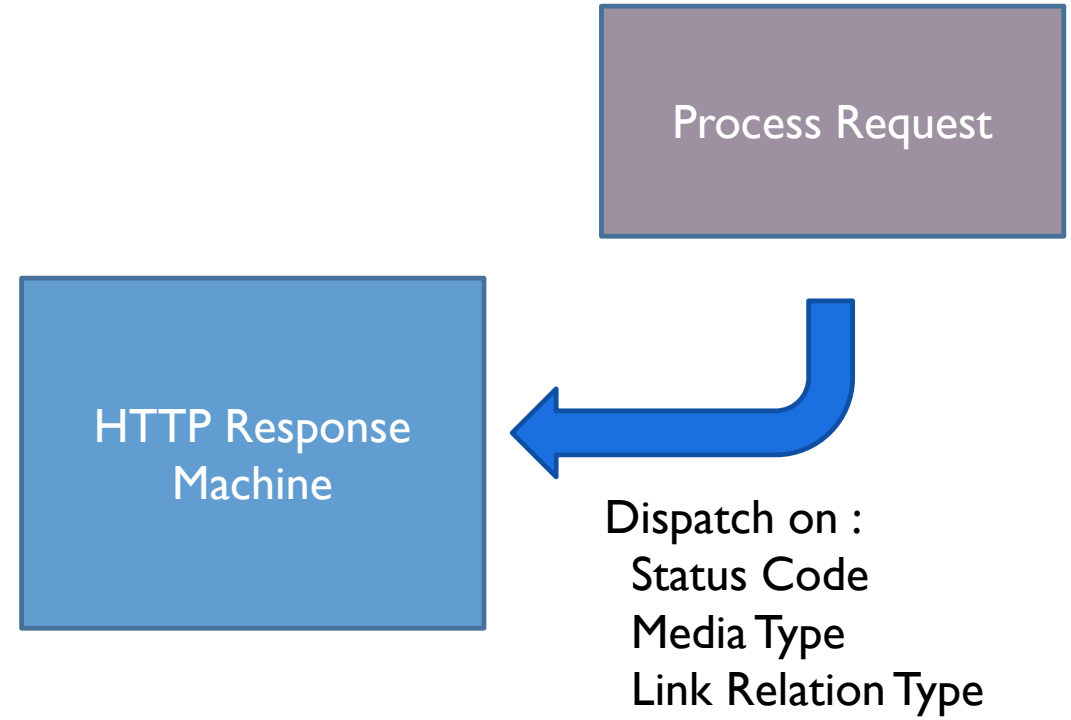
BREAKING REQUEST FROM RESPONSE



CENTRALIZED RESPONSE HANDLING



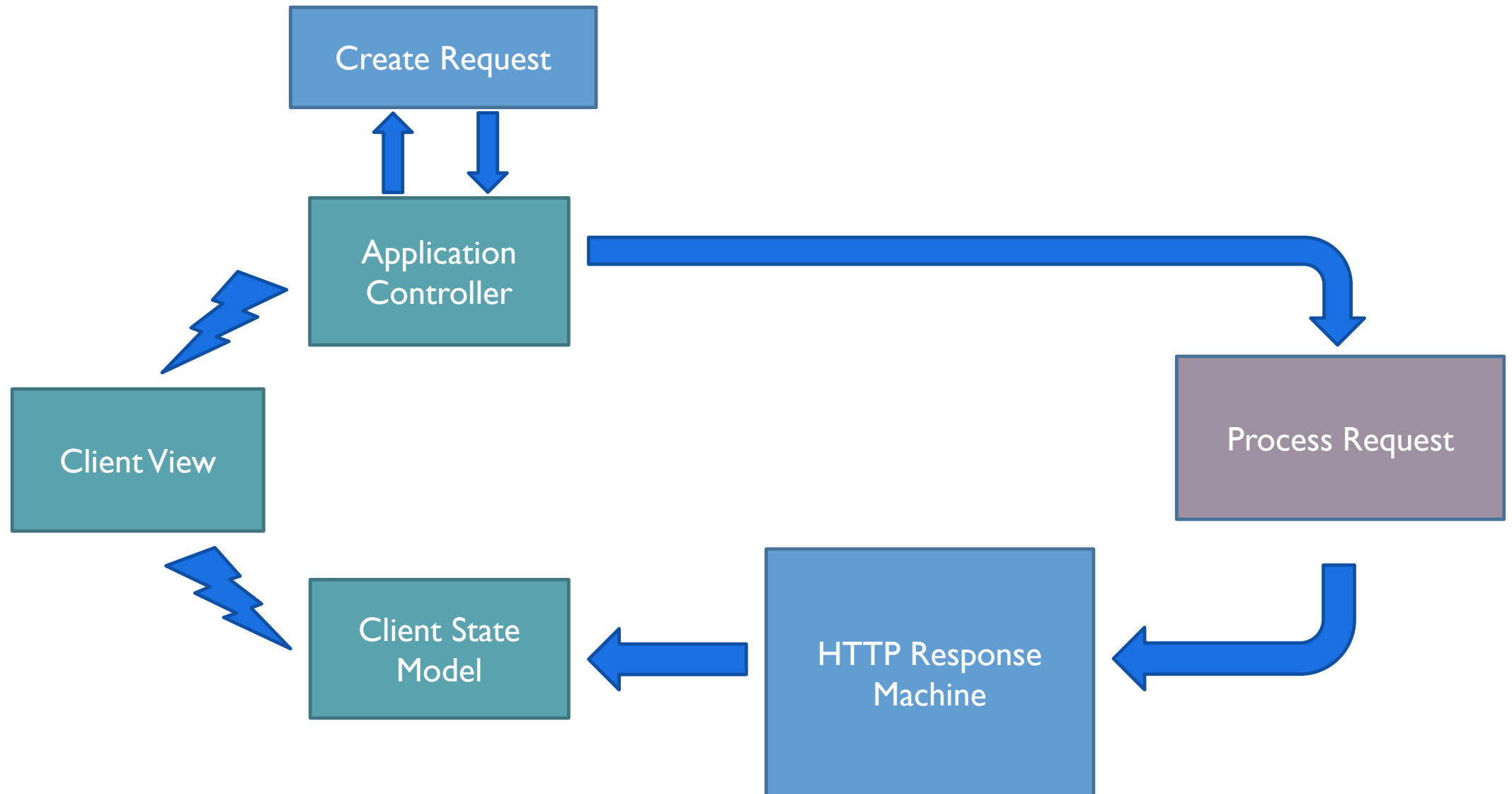
CONTEXT IS KING



CLIENT STATE MANAGEMENT



CLIENT STATE MANAGEMENT



CHEF'S SUGGESTION



BREAKING REQUEST/RESPONSE COUPLING BRINGS FLEXIBILITY

LINK RELATION TYPES ARE YOUR FRIEND

APPLY RESPONSES TO TRANSFORM CLIENT STATE

RELATED TALKS

- Using Hypermedia to avoid the app store –
<https://www.youtube.com/watch?v=LbSM8U2IYkM>
- Crafting Evolvable API Representations –
<https://vimeo.com/131643022>
- Succeeding In Failing –
<https://vimeo.com/131632607>



LIBRARIES



<http://github.com/tavis-software>

Tavis.UriTemplates

Tavis.Link

Tavis.Home

Tavis.Problem

Tavis.JsonPatch

Tavis.HttpCache

Tavis.Auth

Tavis.JsonPointer

Tavis.Hal

Tavis.Status



<http://hapikit.github.io>

Hapikit.net

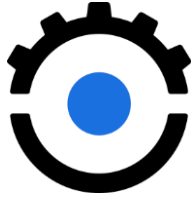
Hapikit.py

Hapikit.go

...

Tooling for building
better HTTP API Client
Libraries

SAMPLES



<https://github.com/Runscope/dotnet-webpack>



<https://github.com/hapikit/github.net.hapikit>



<https://github.com/darrelmiller/ForceLinksForNet>



<https://github.com/hapikit/stormpath.net.hapikit>

';--hibp?

<https://github.com/hapikit/haveibeenpwnd.net.hapikit>



BON APPETIT

Twitter: @darrel_miller

<http://www.bizcoder.com/>

