

SERVERLESS.JS

STEPHEN WHITE

| CALLISTAENTERPRISE.SE

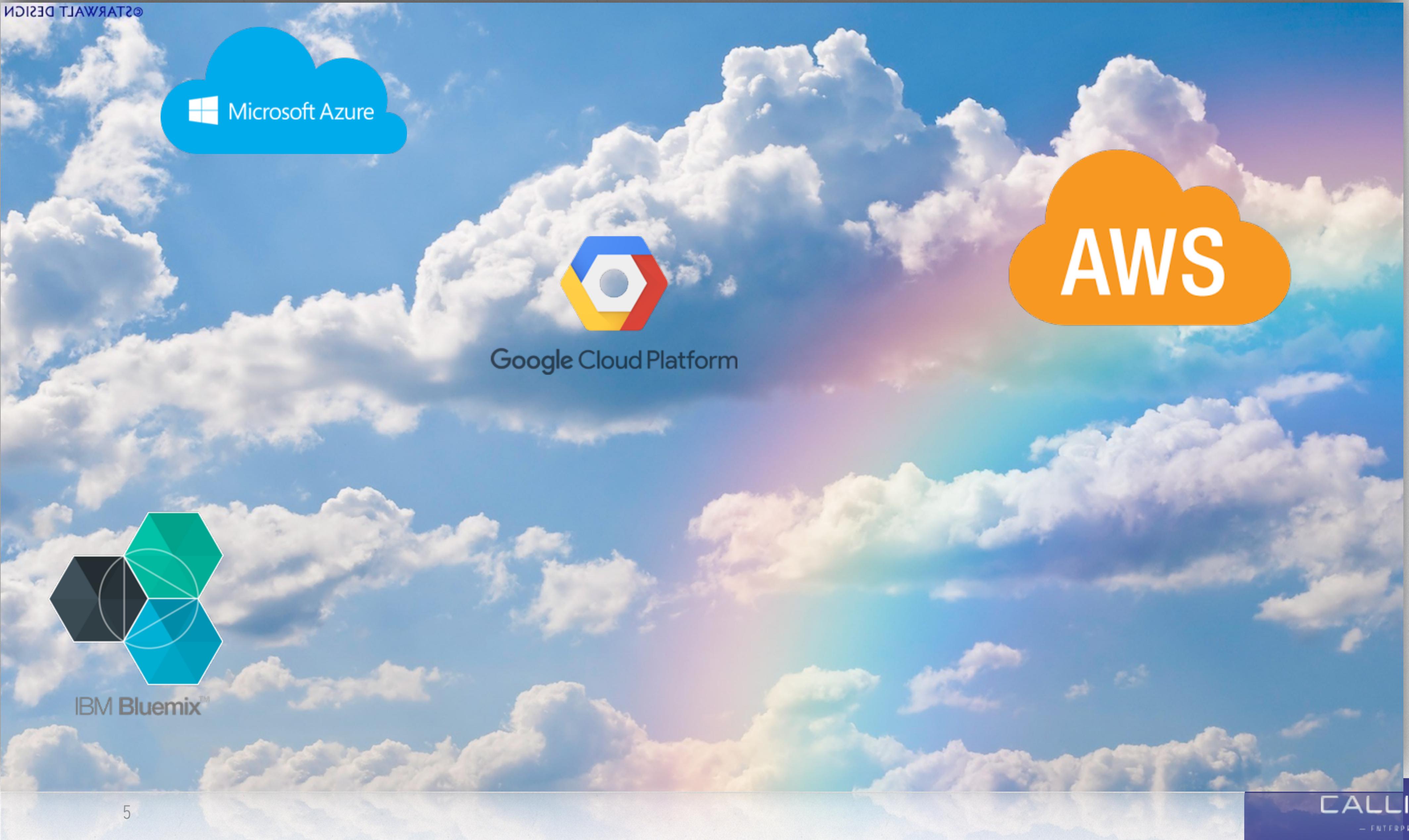
SERVERLESS.JS - AGENDA

- Serverless
- Serverless.JS
 - What
 - How
- Architecture
 - GraphQL
 - Event Driven

Is the absence of a physical server ...

So where does your code go?

SERVERLESS



The Serverless Framework allows you to **deploy auto-scaling, pay-per-execution, event-driven functions** to any **cloud**.

Currently support **Amazon Web Service's Lambda**, and are expanding to support other cloud providers.

Language Support :

- JS
- Java
- Python
- and more (Babel Transpilation Swift ?)

It's a CLI

- serverless create
- serverless deploy
- serverless logs -f users -t log
- serverless webpack serve

- Functions
 - Lambdas
- Events
 - Anything that triggers a lambda
- Resources
 - Infrastructure component
- Services/Stacks
 - Unit of organisation
- Plugins
 - deployment, testing, special resources

SERVERLESS.JS - WHAT - CORE CONCEPTS - FUNCTIONS, RESOURCES STACKS

Services/Stacks



Cloud
Formation



Lambda

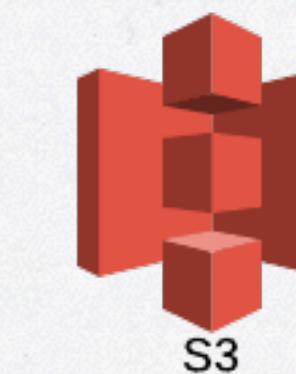
Functions



AmazonAPIGateway



Amazon
CloudFront



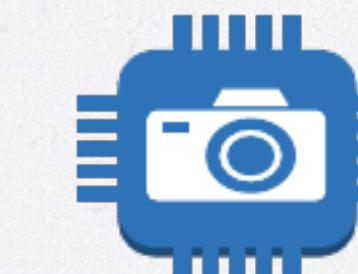
Amazon
DynamoDB



AWS
Cloud



AWSIoT
(topic)



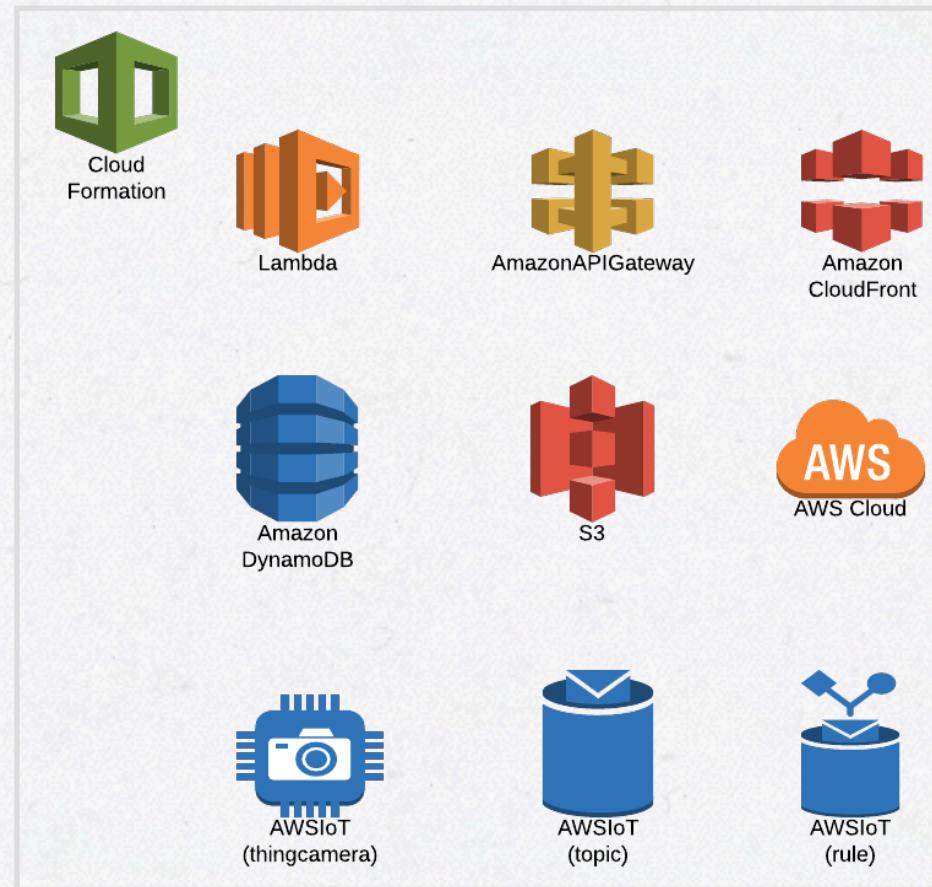
AWSIoT
(thingcamera)



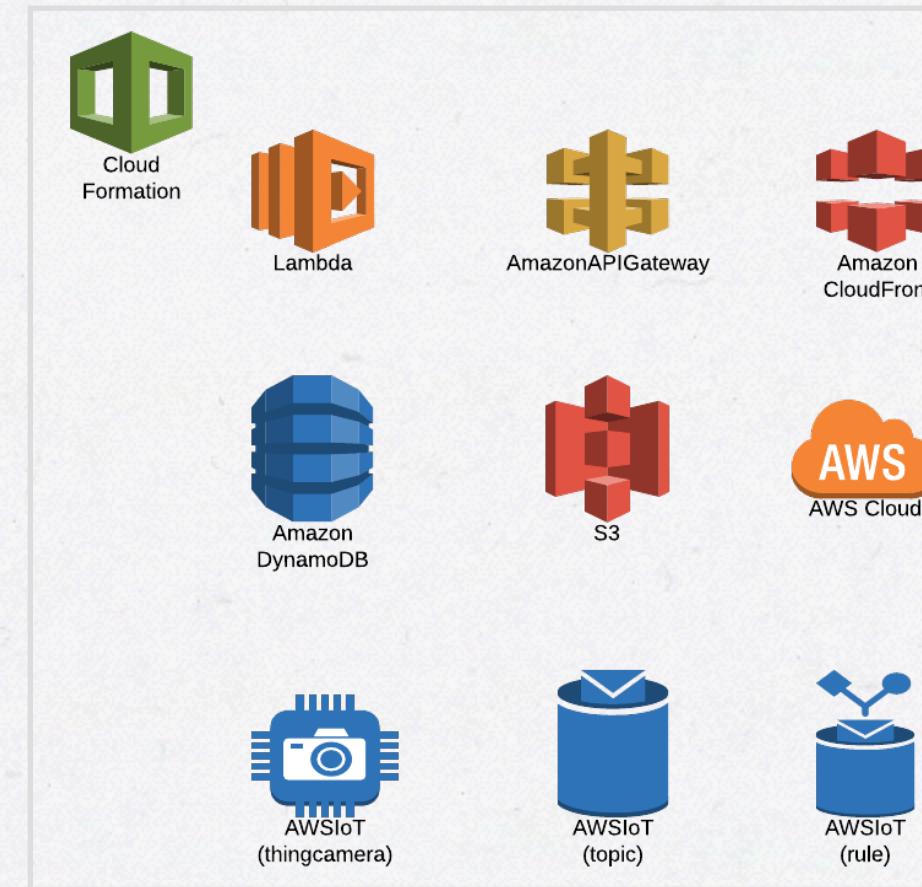
AWSIoT
(rule)

SERVERLESS.JS - HOW - STACKS

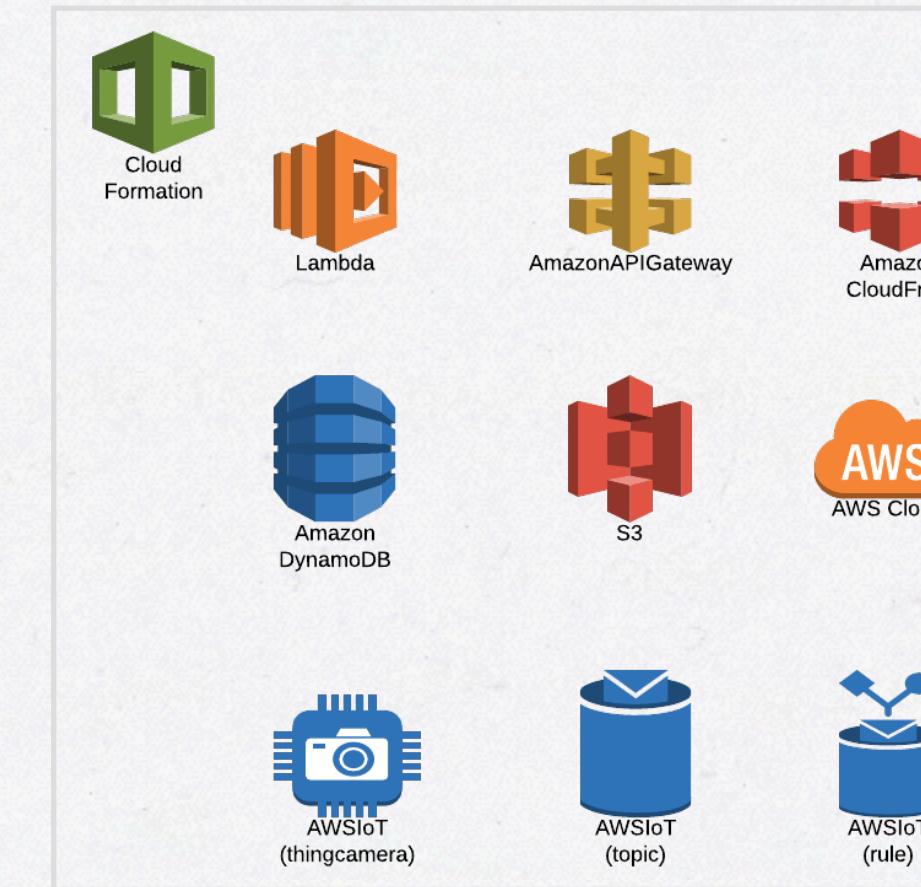
Dev



Test



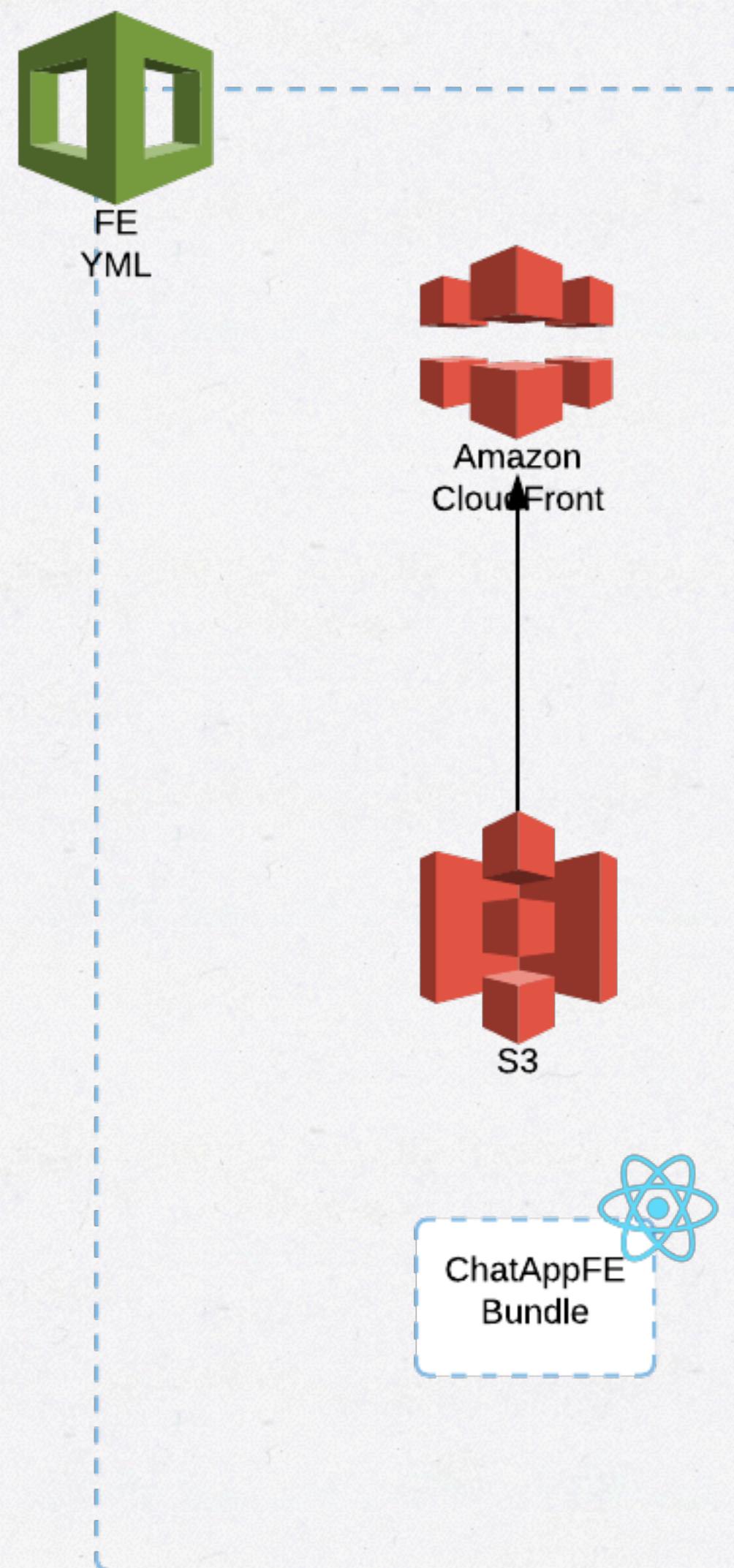
Prod



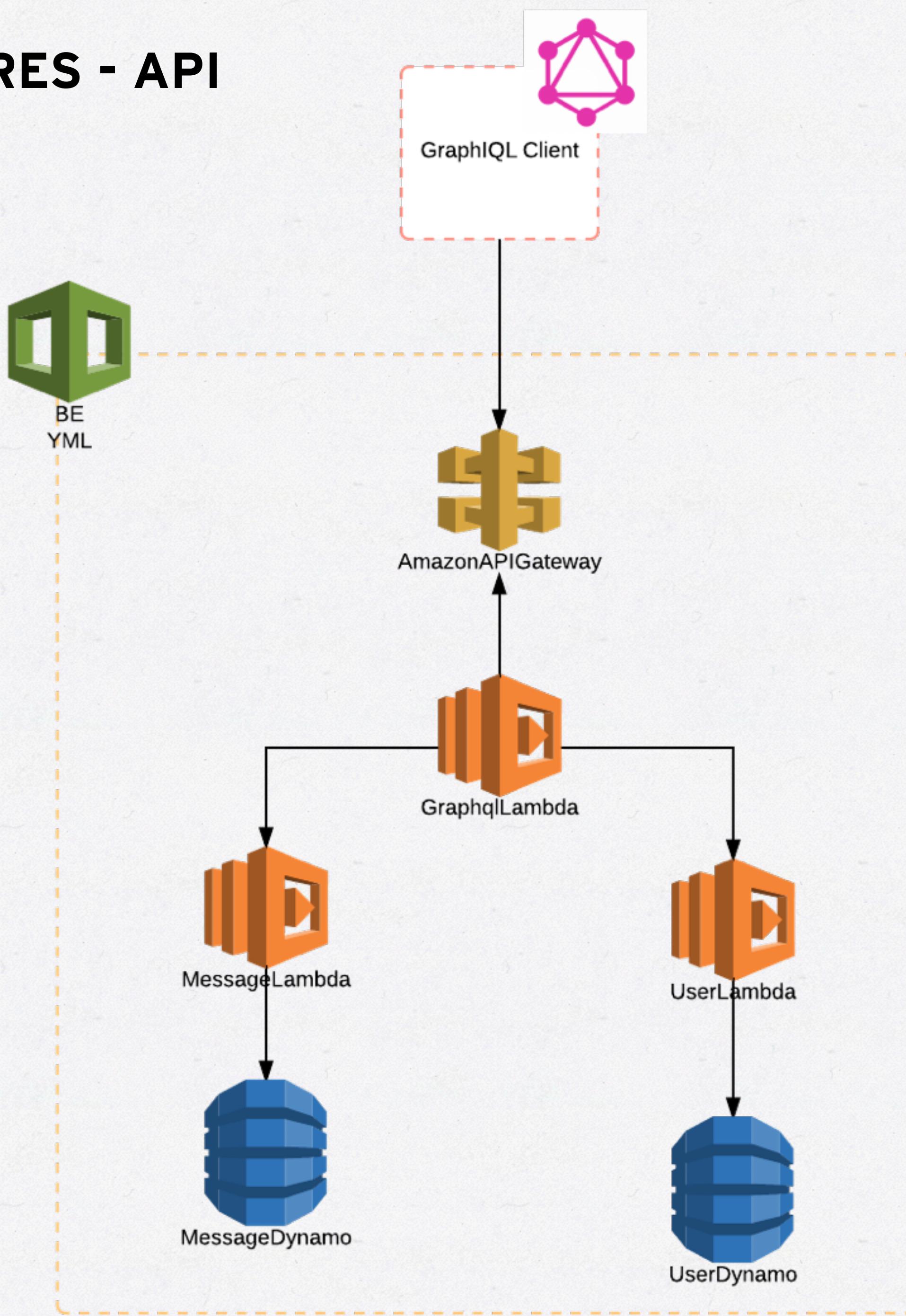
Chat App!

Event Async
API Request/Reply
Frontend End SPA

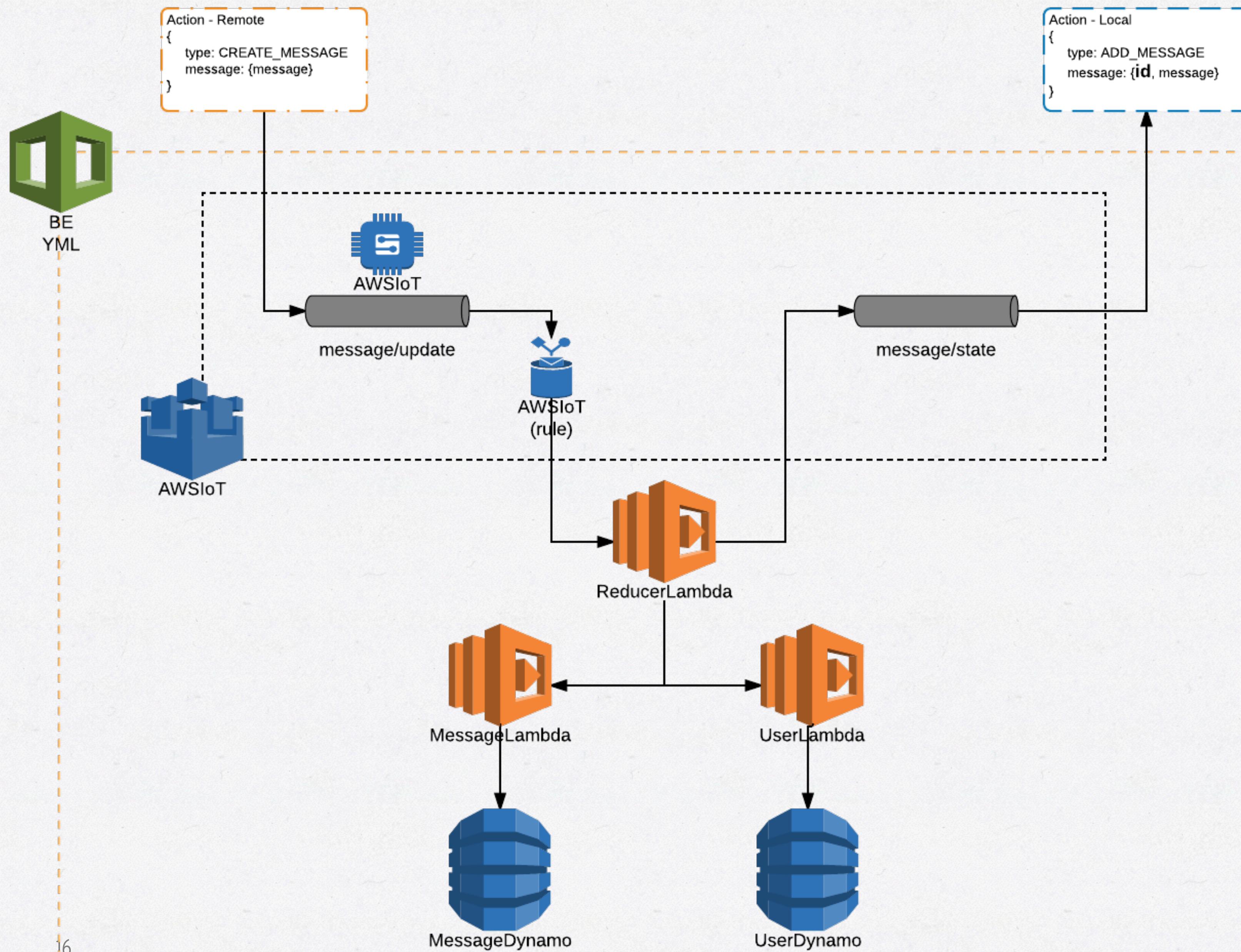
ARCHITECTURE FE



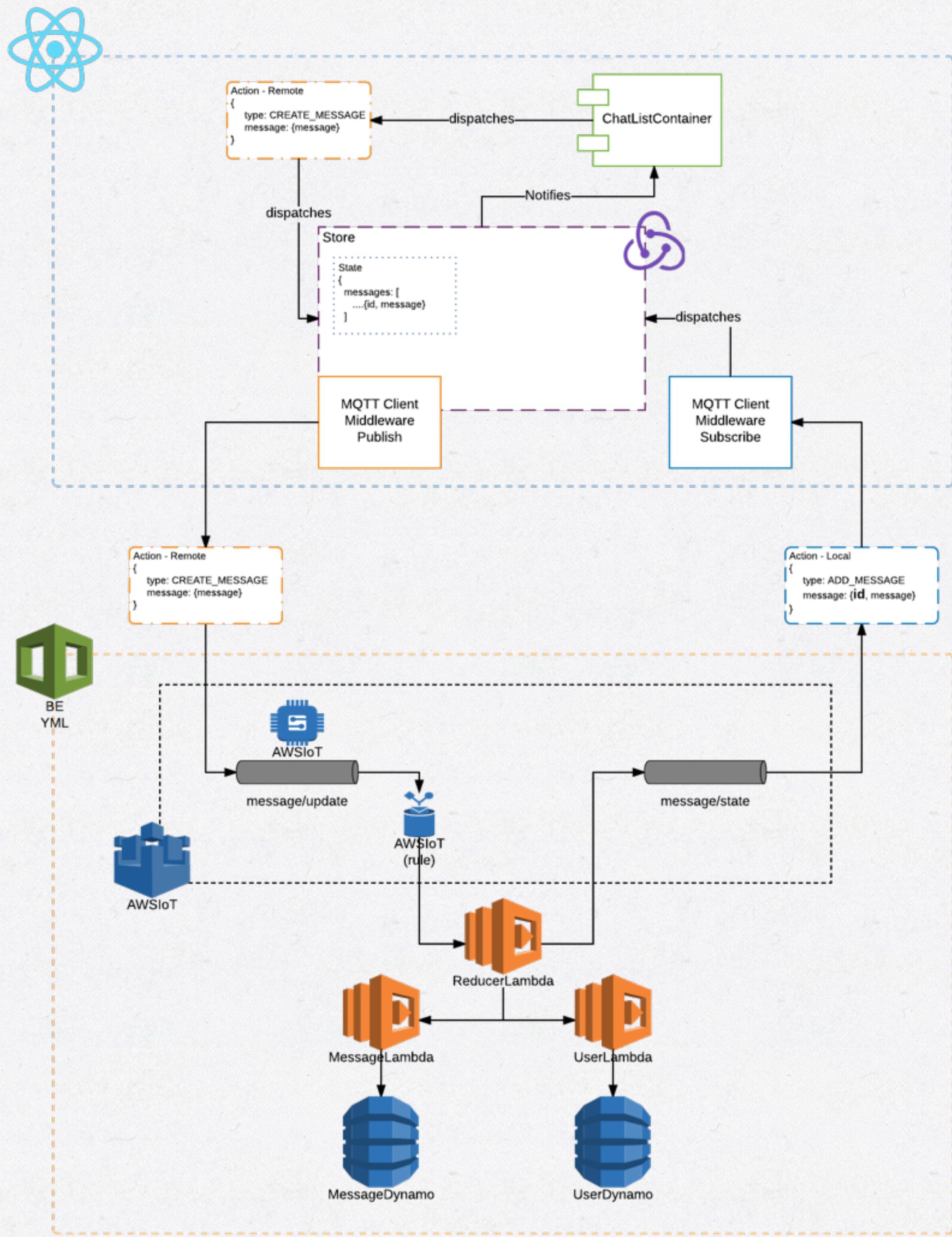
CHAT APP - ARCHITECTURES - API



CHAT APP - ARCHITECTURES - EVENT



CHAT APP



WHAT WE'VE DONE

- Serverless.js - lightning walkthrough
- Deployment
- Logging
- Architecture
 - Backend
 - Event based
 - GraphQL Request/Reply
 - How to do a traditional client server architecture
 - Deployment of Front End (React)

CONCLUSIONS



The Future is Fluffy!

```
serverless create --template aws-nodejs --path myService
```

serverless.yml (Cloud Formation)
handler.js (Lambda)

WHAT IS SERVERLESS - LANGUAGES

	JavaScript /Node	Java	Python	.Net	Swift	Ruby	PHP	C	F#
AWS	X	X	X	X					
AZURE	X	X	X	X			X		X
BLUEMIX	X	X	X	X	X	X	X	X	X
GOOGLE CLOUD	X	X	X	X		X	X		

SERVERLESS.JS - HOW - LAMBDA - HANDLER

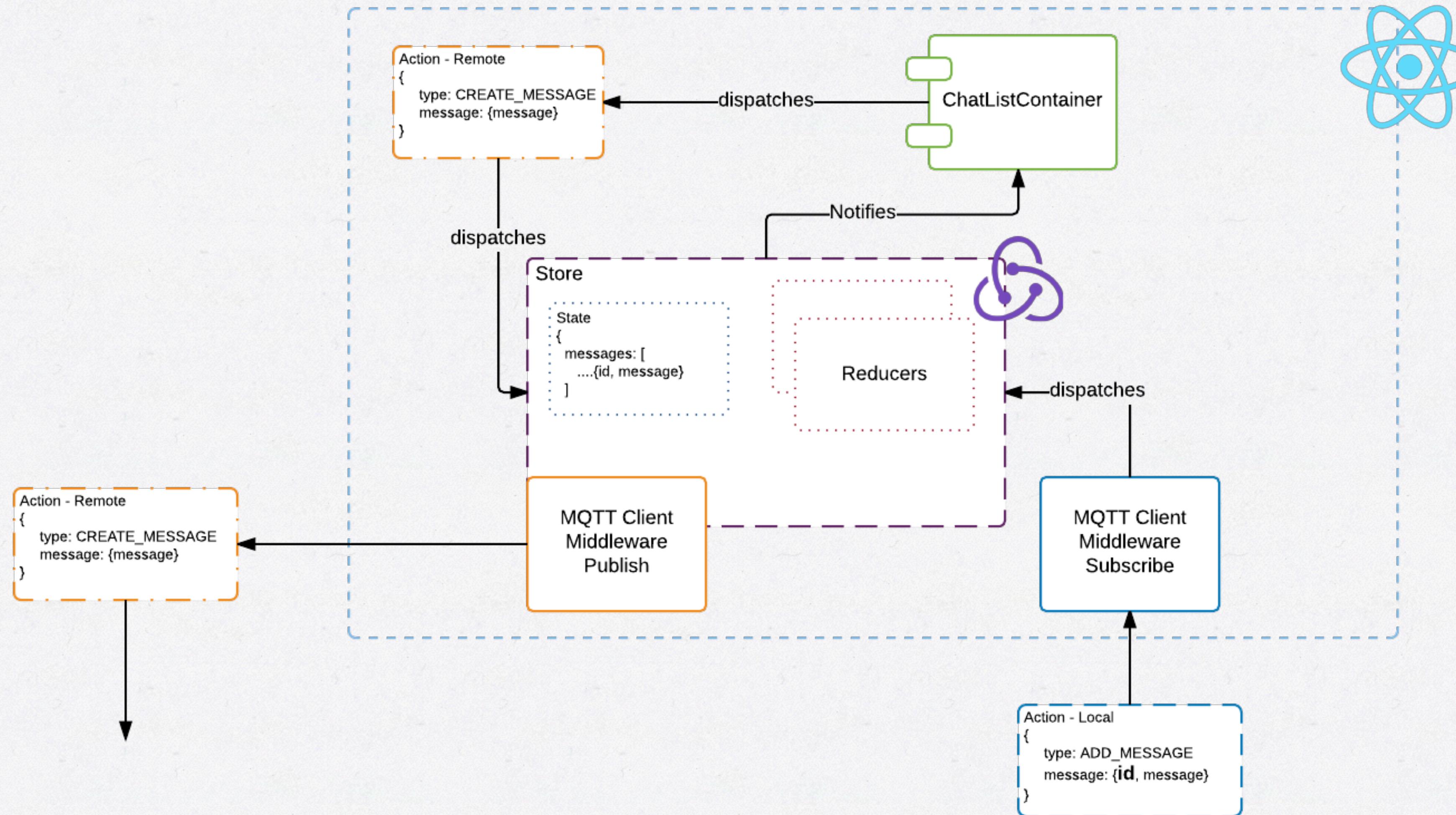
```
function myLambda(event, context, callback)
```

SERVERLESS.JS - HOW- LAMBDA - YML

```
myLambda:  
    handler: index.myLambda  
    environment:  
        IOT_Endpoint: a31ovqfkmg1ev8.iot.eu-west-1.amazonaws.com  
        State_Topic: $aws/things/dev-cadec-messagesX/state  
        Lambdas: "${self:custom.lambda_users}/${self:custom.lambda_messages}"  
    events:  
        - http:  
            path: reducer  
            method: POST  
            integration: lambda  
            cors: true  
        - iot:  
            sql: "SELECT * FROM '$aws/things/dev-cadec-messagesX/shadow/update'"
```

serverless deploy

ARCHITECTURE FRONTEND



REDUX

- Store
 - state, one state tree
- Actions
 - dispatched to store, then passed through all reducers.
- Reducers
 - switch statement, takes actions and mutates state accordingly
- Middleware
 - chance to introspect actions, i.e. remote actions