readme

API Mixtape

SKETCH NOTES

May 17, 2017

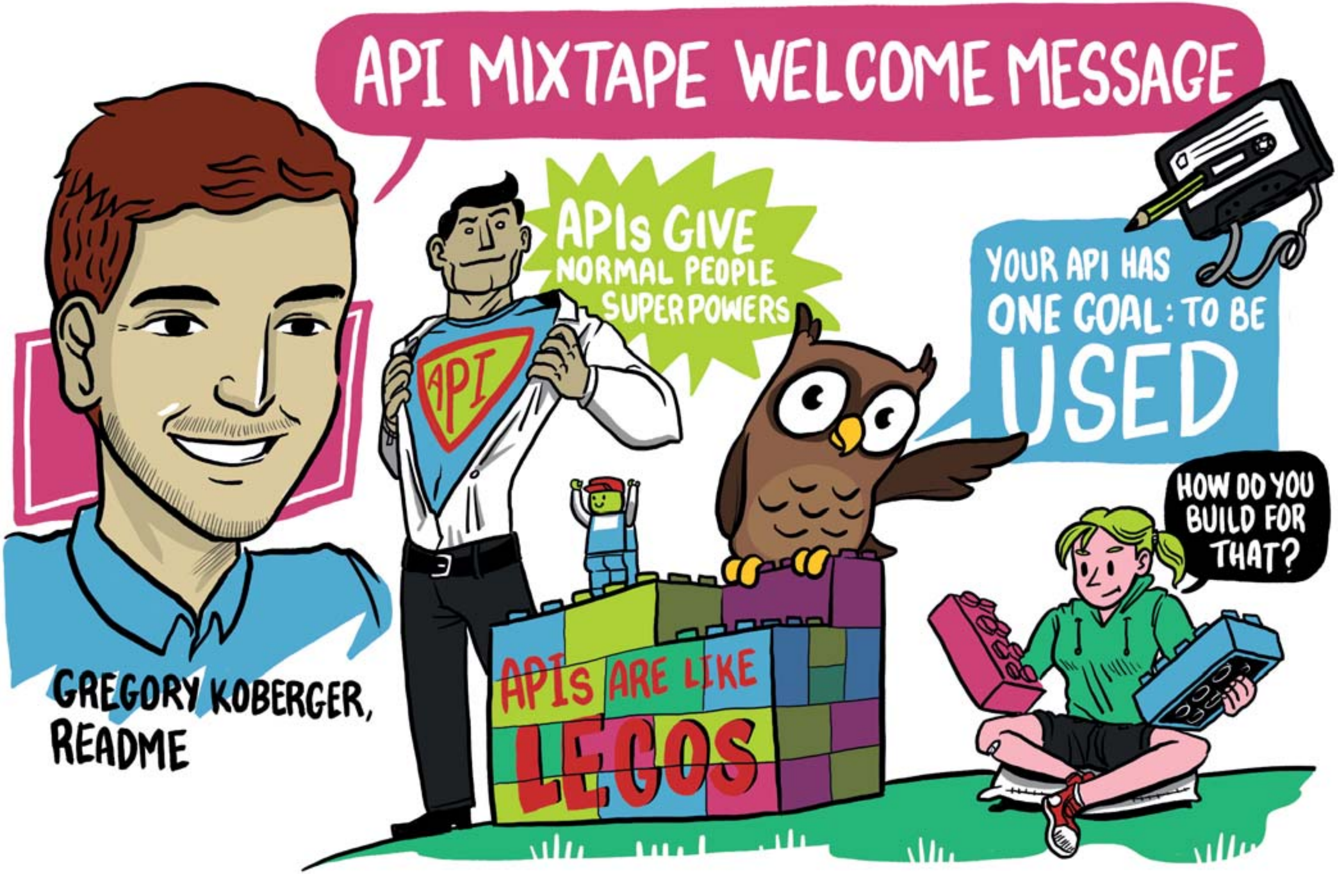
TABLE OF CONTENTS

WELCOME

This book contains sketch notes from API Mixtape, a conference focused on building easy-to-use APIs. You can view the full talks at:

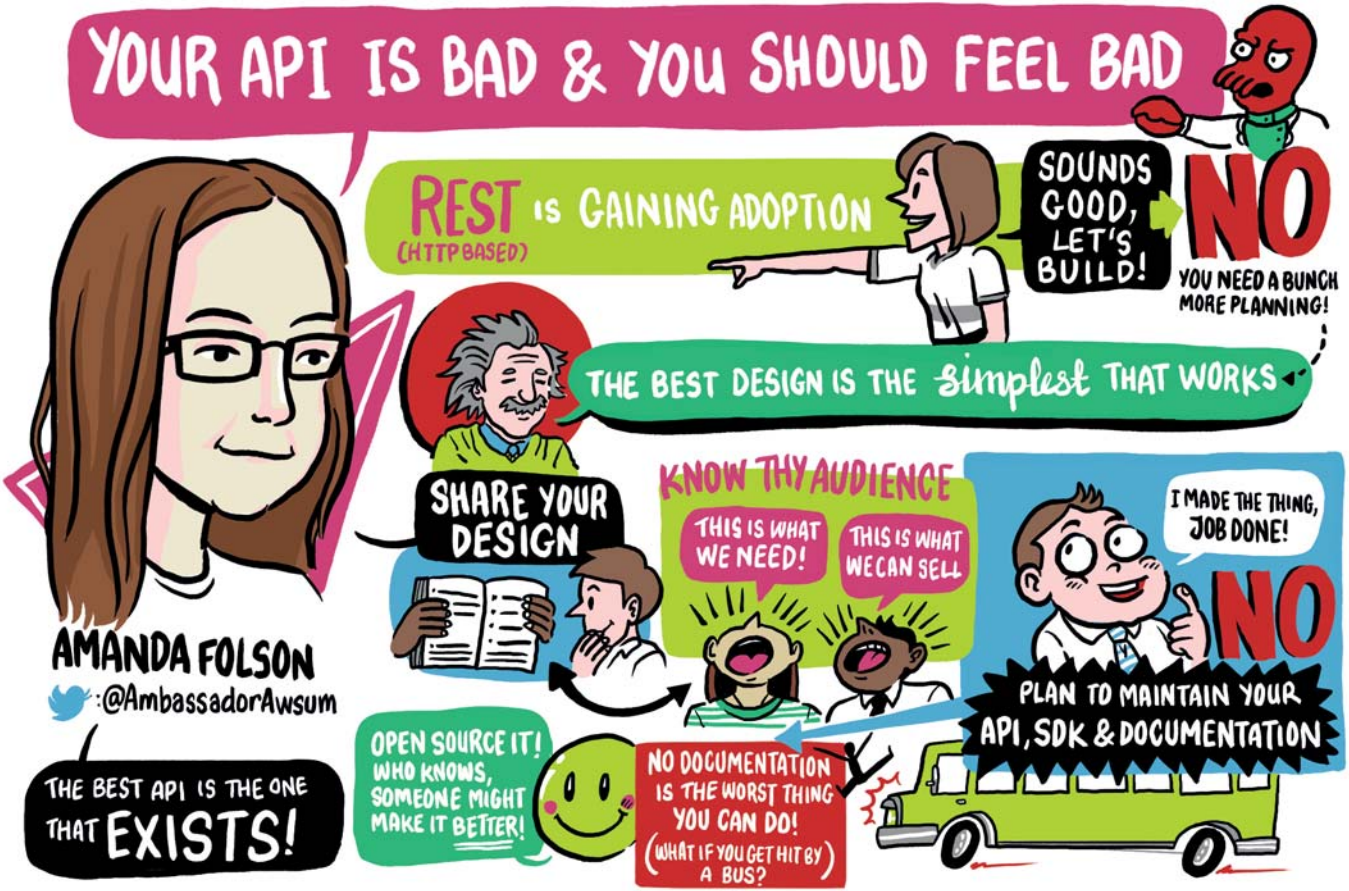
apimixtape.com/talks

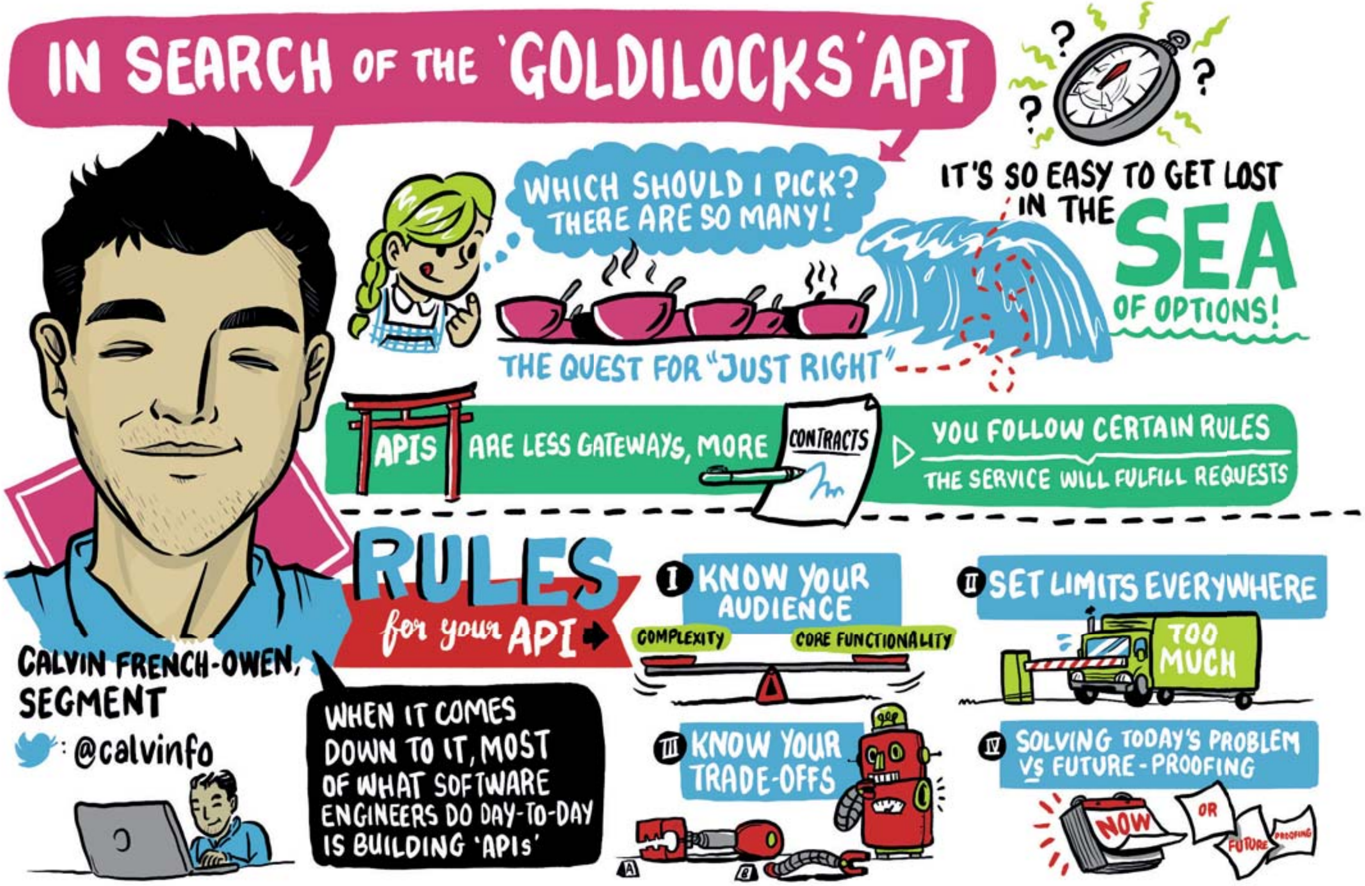
readme

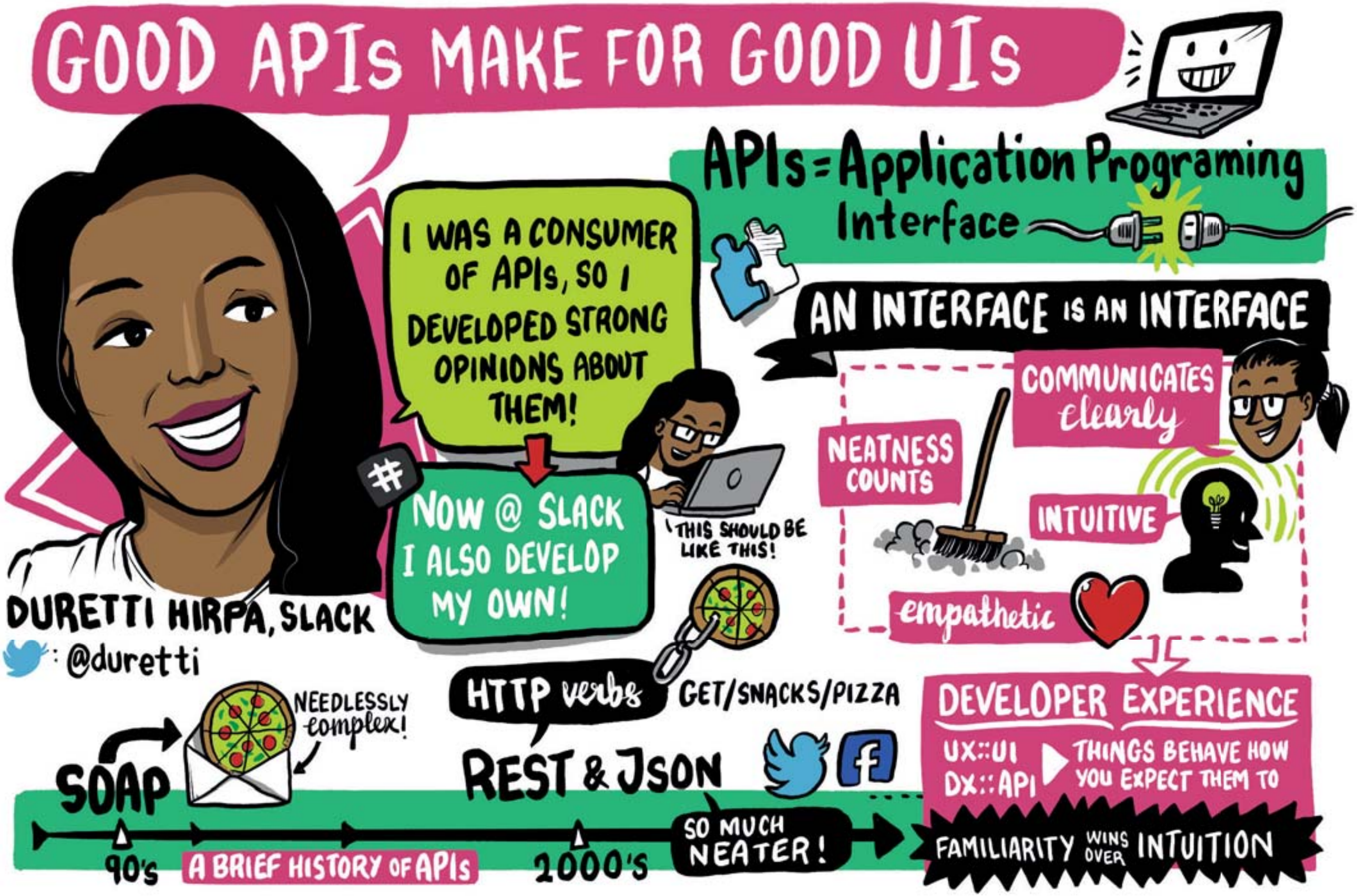

API MIXTAPE WELCOME MESSAGE
APIs GIVE NORMAL PEOPLE SUPERPOWERS
API
YOUR API HAS ONE GOAL: TO BE USED
HOW DO YOU BUILD FOR THAT?
APIs ARE LIKE LEGOS
GREGORY KOBERGER, README

DESIGN

What to think about before you write your first line of code

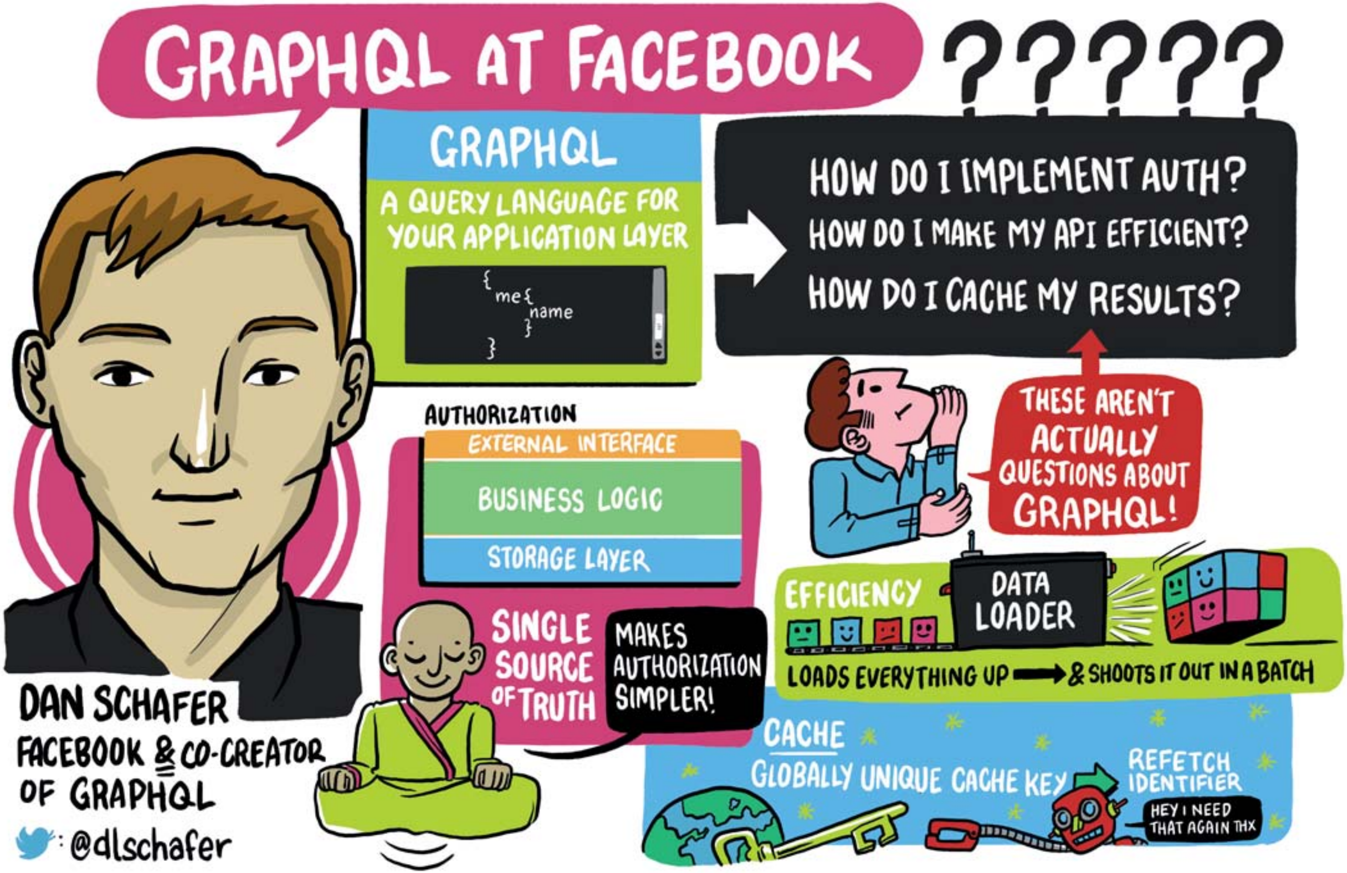

YOUR API IS BAD & YOU SHOULD FEEL BAD
REST (HTTP BASED) IS GAINING ADOPTION
SOUNDS GOOD, LET'S BUILD!
NO
YOU NEED A BUNCH MORE PLANNING!
THE BEST DESIGN IS THE simplest THAT WORKS
SHARE YOUR DESIGN
KNOW THY AUDIENCE
THIS IS WHAT WE NEED!
THIS IS WHAT WE CAN SELL
I MADE THE THING, JOB DONE!
NO
PLAN TO MAINTAIN YOUR API, SDK & DOCUMENTATION
AMANDA FOLSON
:@AmbassadorAwsum
THE BEST API IS THE ONE THAT EXISTS!
OPEN SOURCE IT! WHO KNOWS, SOMEONE MIGHT MAKE IT BETTER!
NO DOCUMENTATION IS THE WORST THING YOU CAN DO! (WHAT IF YOU GET HIT BY A BUS?)


IN SEARCH OF THE 'GOLDILOCKS' API
WHICH SHOULD I PICK? THERE ARE SO MANY!
THE QUEST FOR "JUST RIGHT"
IT'S SO EASY TO GET LOST IN THE SEA OF OPTIONS!
APIS ARE LESS GATEWAYS, MORE CONTRACTS
YOU FOLLOW CERTAIN RULES
THE SERVICE WILL FULFILL REQUESTS
RULES for your API
I KNOW YOUR AUDIENCE
COMPLEXITY
CORE FUNCTIONALITY
II SET LIMITS EVERYWHERE
TOO MUCH
III KNOW YOUR TRADE-OFFS
A
B
IV SOLVING TODAY'S PROBLEM VS FUTURE-PROOFING
NOW
OR
FUTURE PROOFING
CALVIN FRENCH-OWEN, SEGMENT
: @calvinfo
WHEN IT COMES DOWN TO IT, MOST OF WHAT SOFTWARE ENGINEERS DO DAY-TO-DAY IS BUILDING 'APIs'

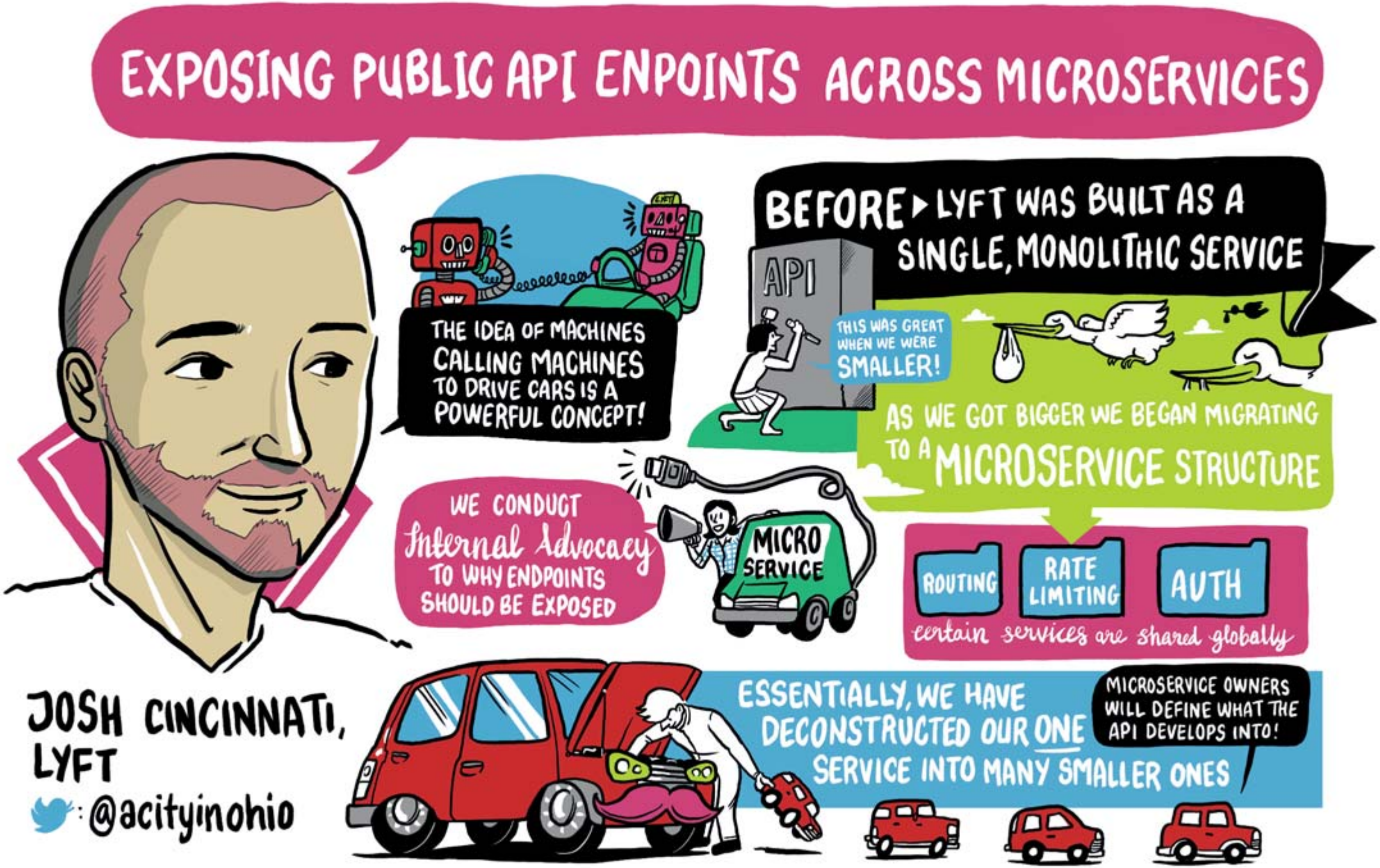

GOOD APIs MAKE FOR GOOD UIs
DURETTI HIRPA, SLACK
: @duretti
I WAS A CONSUMER OF APIs, SO I DEVELOPED STRONG OPINIONS ABOUT THEM!
#
NOW @ SLACK I ALSO DEVELOP MY OWN!
APIs = Application Programing Interface
AN INTERFACE IS AN INTERFACE
COMMUNICATES clearly
NEATNESS COUNTS
INTUITIVE
empathetic
'THIS SHOULD BE LIKE THIS!
HTTP verbs
GET/SNACKS/PIZZA
SOAP
NEEDLESSLY complex!
REST & JSON
SO MUCH NEATER!
DEVELOPER EXPERIENCE
UX::UI
DX::API
THINGS BEHAVE HOW YOU EXPECT THEM TO
FAMILIARITY WINS OVER INTUITION
90's
A BRIEF HISTORY OF APIs
2000's

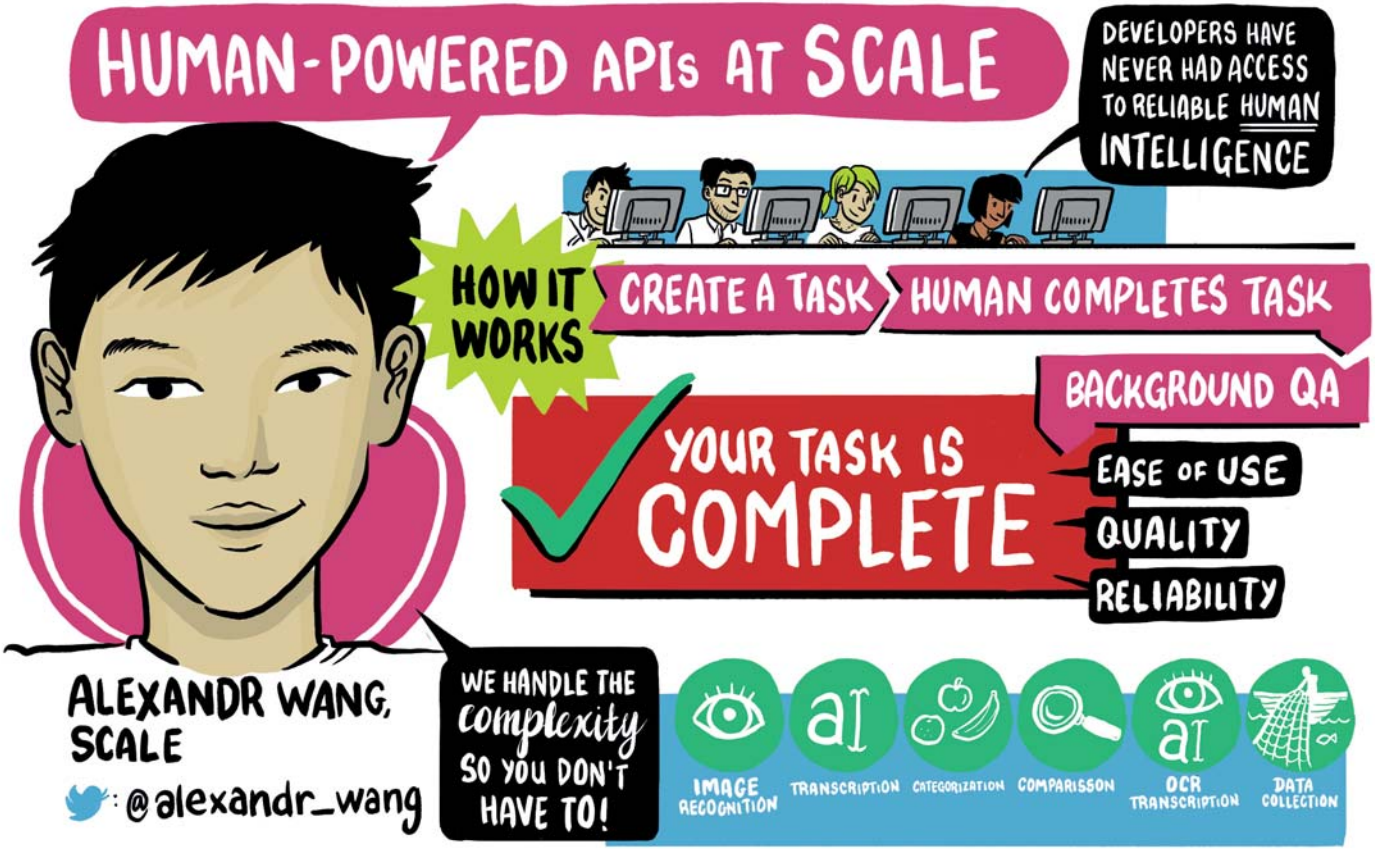
BUILD

Everything is designed, so it’s
time to get technical!


GRAPHQL AT FACEBOOK
?????
GRAPHQL
A QUERY LANGUAGE FOR YOUR APPLICATION LAYER
HOW DO I IMPLEMENT AUTH?
HOW DO I MAKE MY API EFFICIENT?
HOW DO I CACHE MY RESULTS?
THESE AREN'T ACTUALLY QUESTIONS ABOUT GRAPHQL!
AUTHORIZATION
EXTERNAL INTERFACE
BUSINESS LOGIC
STORAGE LAYER
SINGLE SOURCE OF TRUTH
MAKES AUTHORIZATION SIMPLER!
EFFICIENCY
DATA LOADER
LOADS EVERYTHING UP → & SHOOTS IT OUT IN A BATCH
CACHE
GLOBALLY UNIQUE CACHE KEY
REFETCH IDENTIFIER
HEY I NEED THAT AGAIN THX
DAN SCHAFER
FACEBOOK & CO-CREATOR OF GRAPHQL
: @dlschafer

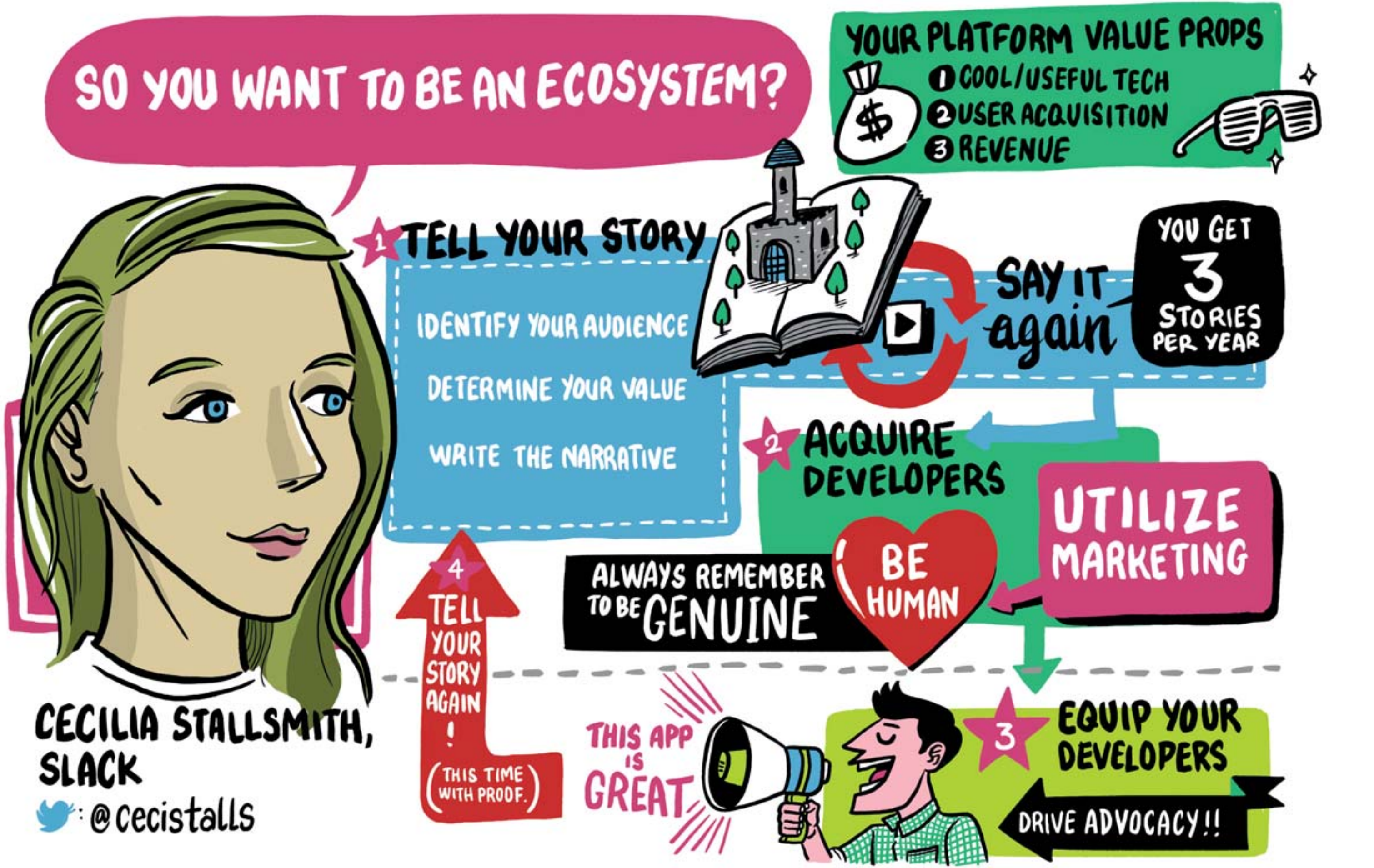
```
{
  me{
    name
  }
}
```


EXPOSING PUBLIC API ENPOINTS ACROSS MICROSERVICES
THE IDEA OF MACHINES CALLING MACHINES TO DRIVE CARS IS A POWERFUL CONCEPT!
LYFT
BEFORE ▸ LYFT WAS BUILT AS A SINGLE, MONOLITHIC SERVICE
API
THIS WAS GREAT WHEN WE WERE SMALLER!
AS WE GOT BIGGER WE BEGAN MIGRATING TO A MICROSERVICE STRUCTURE
WE CONDUCT Internal Advocacy TO WHY ENDPOINTS SHOULD BE EXPOSED
MICRO SERVICE
ROUTING
RATE LIMITING
AUTH
certain services are shared globally
JOSH CINCINNATI, LYFT
: @acityinohio
ESSENTIALLY, WE HAVE DECONSTRUCTED OUR ONE SERVICE INTO MANY SMALLER ONES
MICROSERVICE OWNERS WILL DEFINE WHAT THE API DEVELOPS INTO!


HUMAN-POWERED APIs AT SCALE
DEVELOPERS HAVE NEVER HAD ACCESS TO RELIABLE HUMAN INTELLIGENCE
HOW IT WORKS
CREATE A TASK
HUMAN COMPLETES TASK
BACKGROUND QA
YOUR TASK IS COMPLETE
EASE OF USE
QUALITY
RELIABILITY
ALEXANDR WANG, SCALE
: @alexandr_wang
WE HANDLE THE complexity SO YOU DON'T HAVE TO!
IMAGE RECOGNITION
aI
TRANSCRIPTION
CATEGORIZATION
COMPARISSON
aI
OCR TRANSCRIPTION
DATA COLLECTION

MARKET

Build it and they will... probably not come.
You need to get the word out!


SO YOU WANT TO BE AN ECOSYSTEM?
CECILIA STALLSMITH,
SLACK
: @cecistalls
YOUR PLATFORM VALUE PROPS
$
1 COOL/USEFUL TECH
2 USER ACQUISITION
3 REVENUE
1 TELL YOUR STORY
IDENTIFY YOUR AUDIENCE
DETERMINE YOUR VALUE
WRITE THE NARRATIVE
SAY IT again
YOU GET 3 STORIES PER YEAR
2 ACQUIRE DEVELOPERS
UTILIZE MARKETING
BE HUMAN
ALWAYS REMEMBER TO BE GENUINE
4 TELL YOUR STORY AGAIN !
(THIS TIME WITH PROOF.)
THIS APP IS GREAT
3 EQUIP YOUR DEVELOPERS
DRIVE ADVOCACY!!

readme

We’re launching a new product called **ReadMe Build**!

Great documentation is only possible if there’s a close integration with the API. We’re creating Build to make it easy to build world-class APIs in minutes, and make it dead simple for the end user.

https://readme.build

Introducing...

readme BUILD

AN EASY WAY TO BUILD & CONSUME AWESOME APIs

CREATE AN API IN 3 MINUTES!

```
$ npm install api -g
```

1) INSTALL THE API TOOL

```
$ api init
```

2) CREATE A NEW API

3) EDIT THE API FILE

```
$ api deploy
```

4) DEPLOY THE API

DONE

CONSUME AN API

USE YOUR OWN MICROSERVICES, OR CONSUME PUBLIC APIs FROM THE REGISTRY

```
const api = require('api')('prod_gkob_8Jalja181nLJadJpqqz');
api('math').do('multiply', {
  x: 1, y:5
}, function(err, results, meta) {
  console.log(results); });
```

NO URLS, API METHODS, HEADERS, TOKENS, MIME TYPES,
IT JUST WORKS!!

FEATURES

- ★ VERSIONING
- ★ LOGS
- ★ RATE LIMITING
- ★ DISCOVERY
- ★ PAYMENTS
- ★ AUTHENTICATION
- ★ INTERACTIVE DOCUMENTATION