
Image: wateroakforge.com

TURNING SYSADMINS INTO CLOUD PLATFORM ENGINEERS

UCCSC 2018

HI!
MY NAME IS

Cliff Pearson

What's a "Platform Engineer"?

PROFESSION RENAMING COMPETITION

- DevOps (Amazon)
- SRE (Google)
- NoOps (The Deluded)

As a platform engineer, your job is to make sure all the components work together as a single, integrated whole. This is a different role then a DevOps engineer; you are responsible for the infrastructure of the platform, not any specific application in that platform.

SO IT MAKES SENSE…

- The term actually most closely reflects what we do
- Wrangling hardware, installer DVDs and cables has been nostalgia for over a decade
- Once you manage enough systems deterministic builds becomes a survival mechanism
- Our team has works closely with application teams to build a common platform for years

WHAT'S CHANGED

- Virtualization stops being about hardware consolidation and starts being about self-service
- VMs are totally disposable
- Sysadmins are no longer gatekeepers
- Everything is now an API. Developers love APIs

DEV VS. SYS… ROUND ONE

- Great developers solve for the applications and service (narrowly scoped)

- Great sysadmins solve for reliable and reproducible infrastructure (broadly scoped)

- This is a natural and healthy tension

spreadsheet.com

Warner Brothers

WHAT DOES IT MEAN?

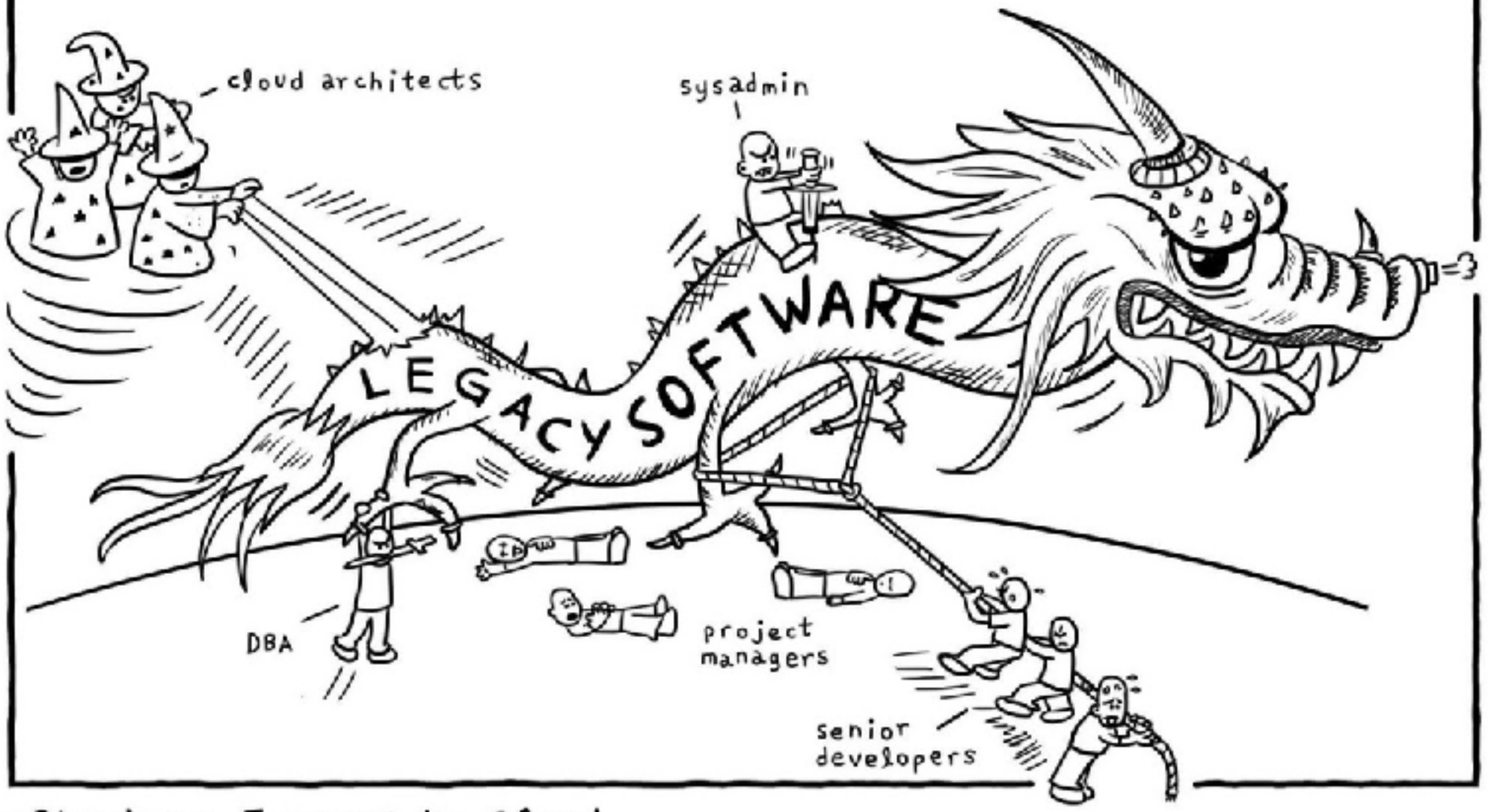
- Keeping thins loose kinda works
- Drawing strict boundaries kinda works
- Not doing anything confuses everyone
- Starting with requirements really helps

OUR EXPERIENCE

- Drop-dead date on when we have to be "done"
- Lock engineers in a room and clear their schedule
- Buy them two pizzas
- Accept big compromises
- PROMISE engineers that the compromises will be corrected (and mean it)

It is often easier to move your application to the cloud
than it is to manage it once it's there

The Enterprise Journey to Cloud



Startups Journey to Cloud

Daniel Stori {turnoff.us}
Thanks to Michael Tharrington

HEROISM IS NOT A SUSTAINABLE MODEL

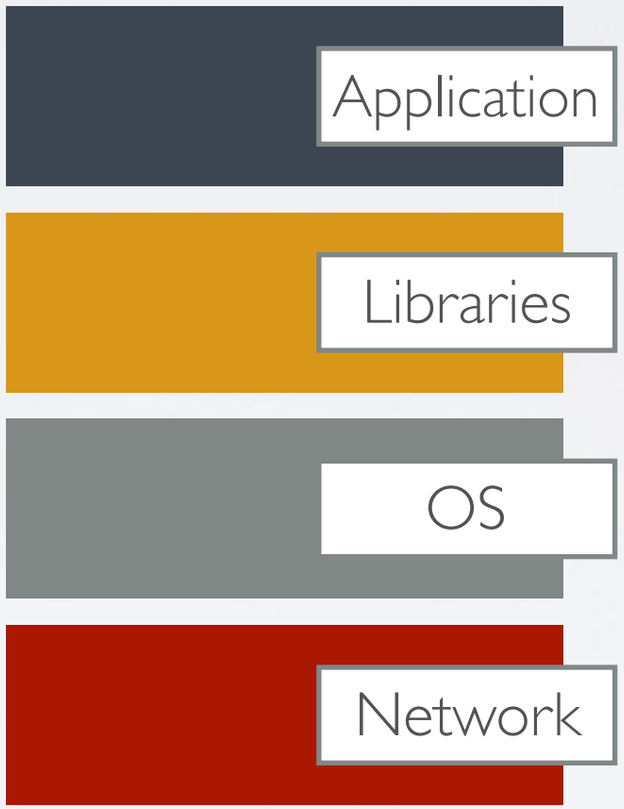
- These people burn out
- Those not involved don't learn, becoming more disaffected
- They build things in a vacuum
- Lack of communications creates de-facto vertical silos with almost impenetrable boundaries
- NIH syndrome on each team


IT'S A
TRAP

"Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations."

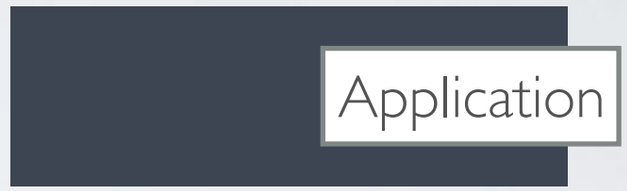
–Melvin Conway

BASIC APP STACK



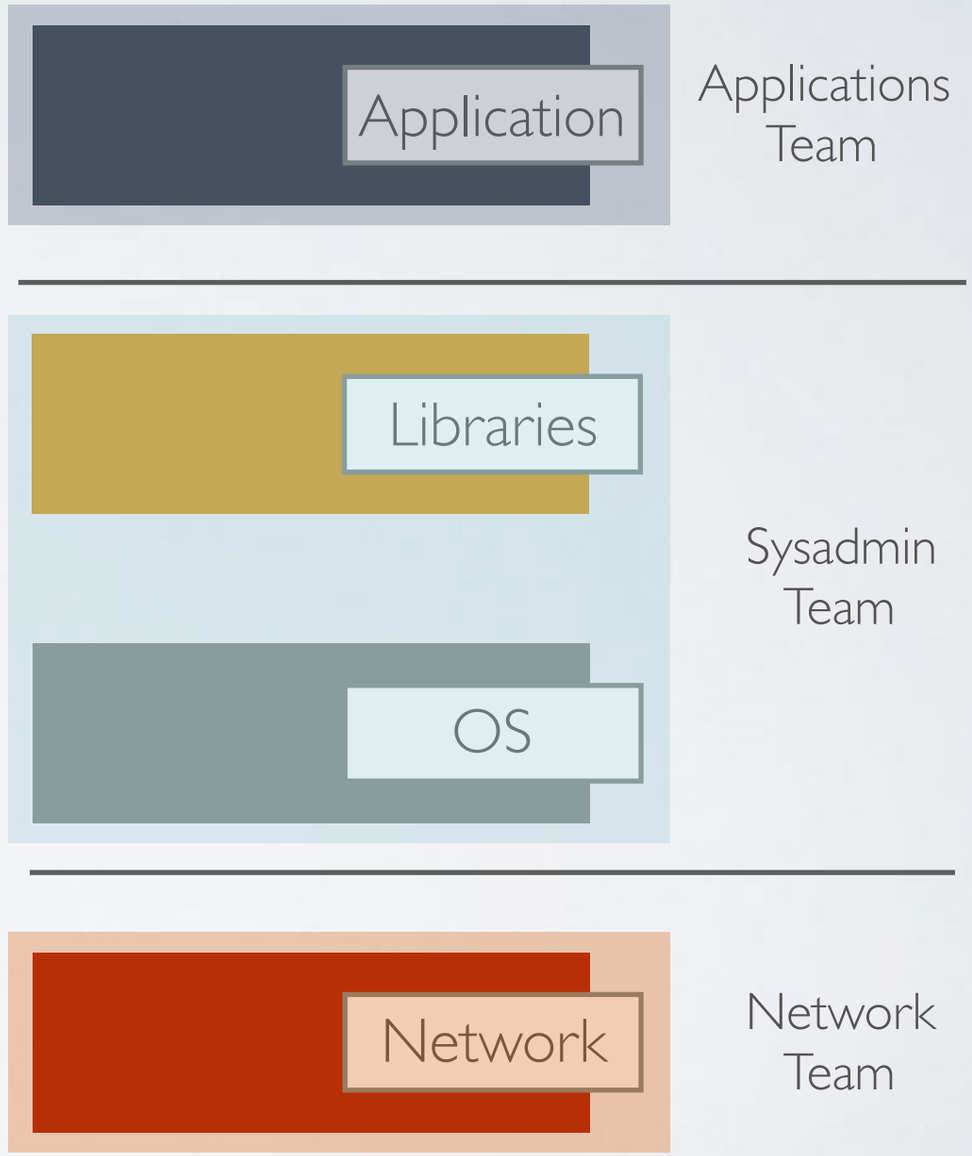
CLASSIC PATTERN

- VM as a first-class citizen
- Loose coupling between system and application
- Build it, manage it, patch it, love it



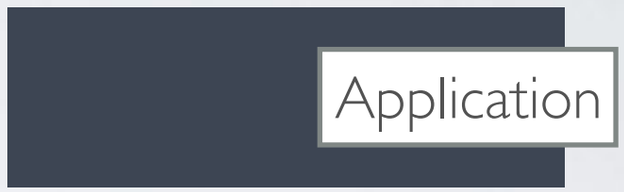
WHAT WOULD CONWAY PREDICT?

- A "department" style org
  - Network Department
  - Systems Department
  - Application Department
  - Database Department
  - Storage Department
  - Etc…



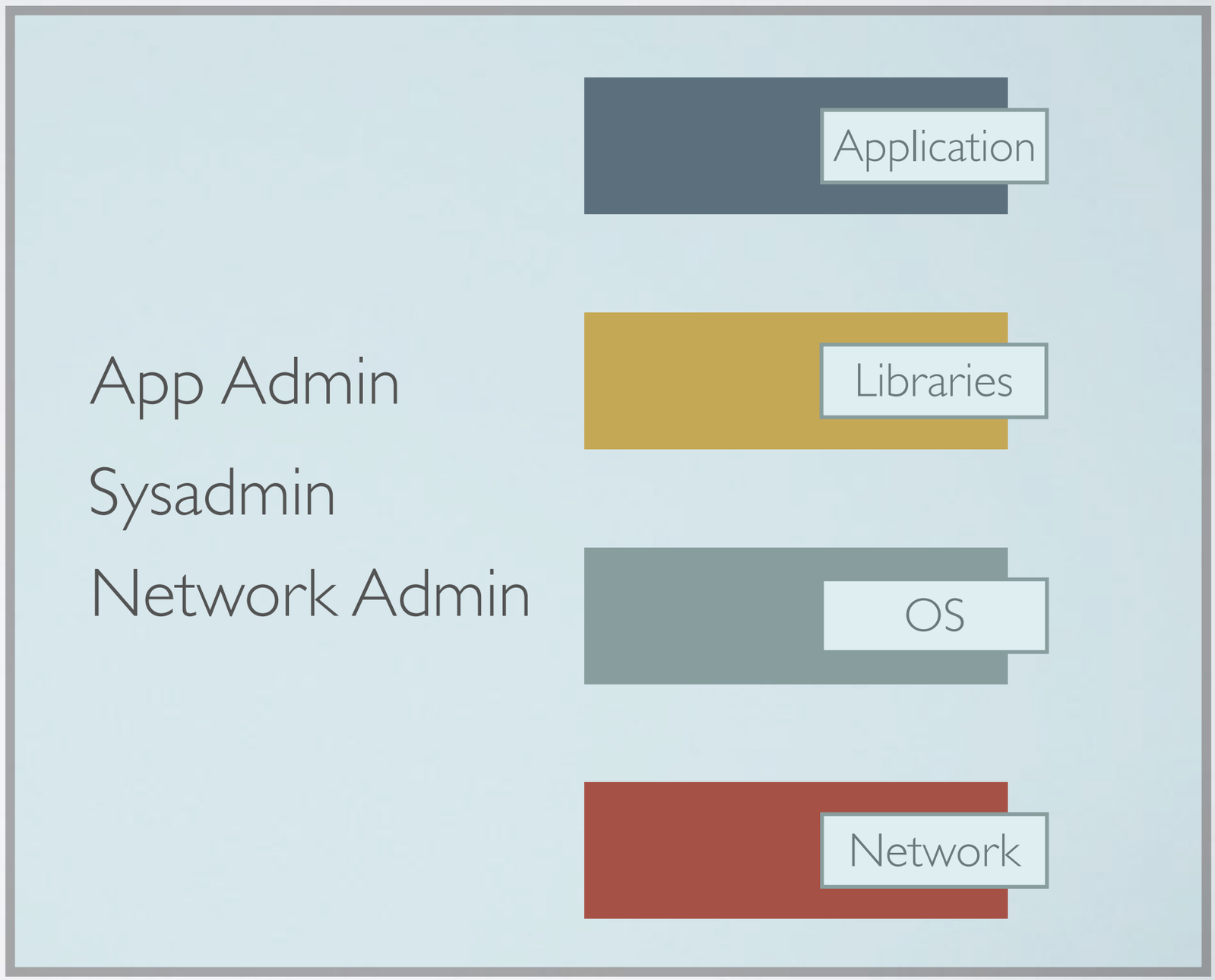
TWO-PIZZA MIGRATION

- Built by tiger team under time pressure
- Tight coupling between system and application
- Made from templates and recipes
- Rebuild to upgrade



WHAT WOULD CONWAY PREDICT?

A vertical silo specific to each application



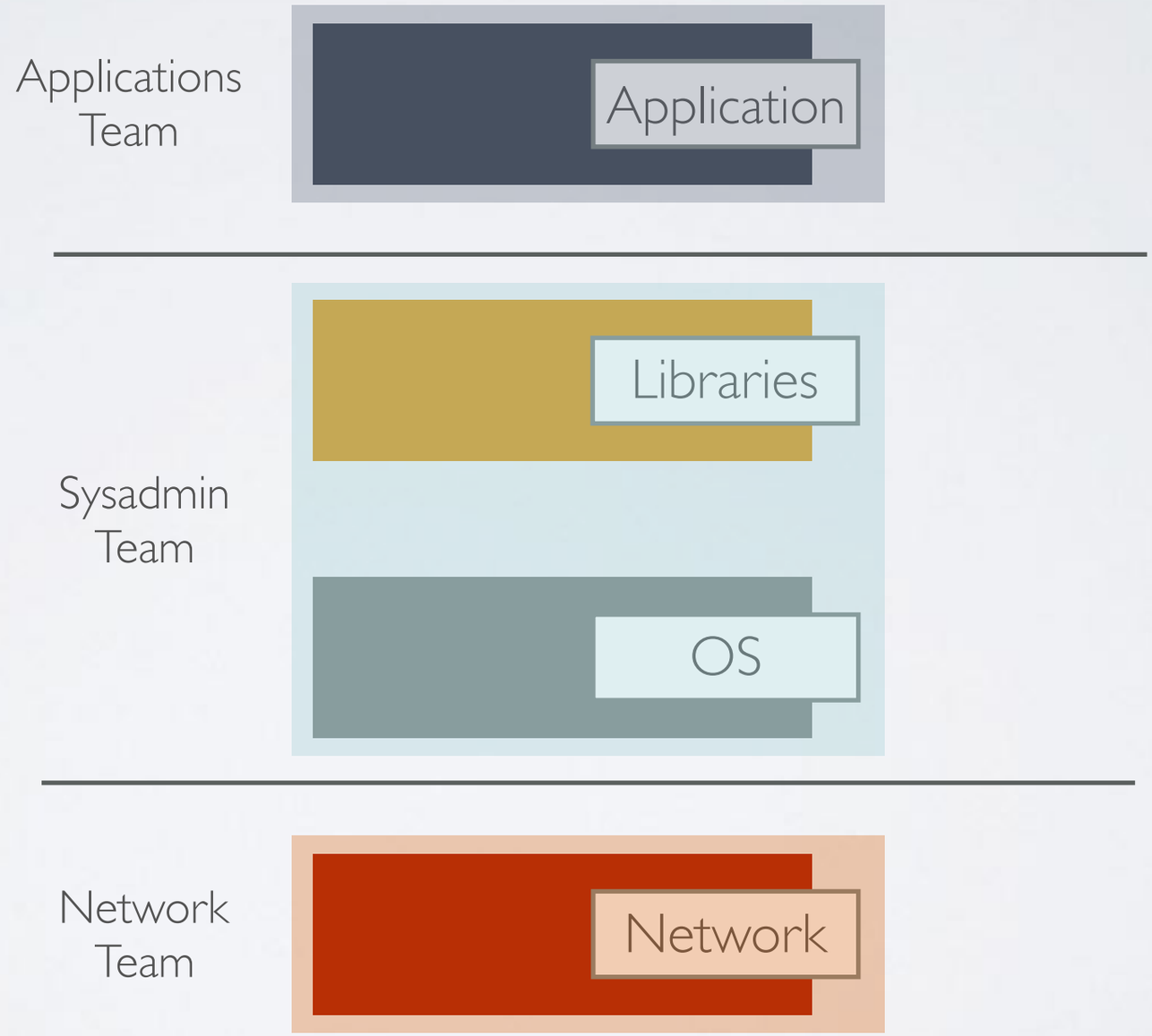
IT MOVED FAST BUT…

- Everyone on that team made a million decisions that tightly coupled every line of code to that specific application
- Their work is not particularly usable by other teams
- Their knowledge extends only to that specific application
- Downstream integrations become bespoke (security, logging, etc)

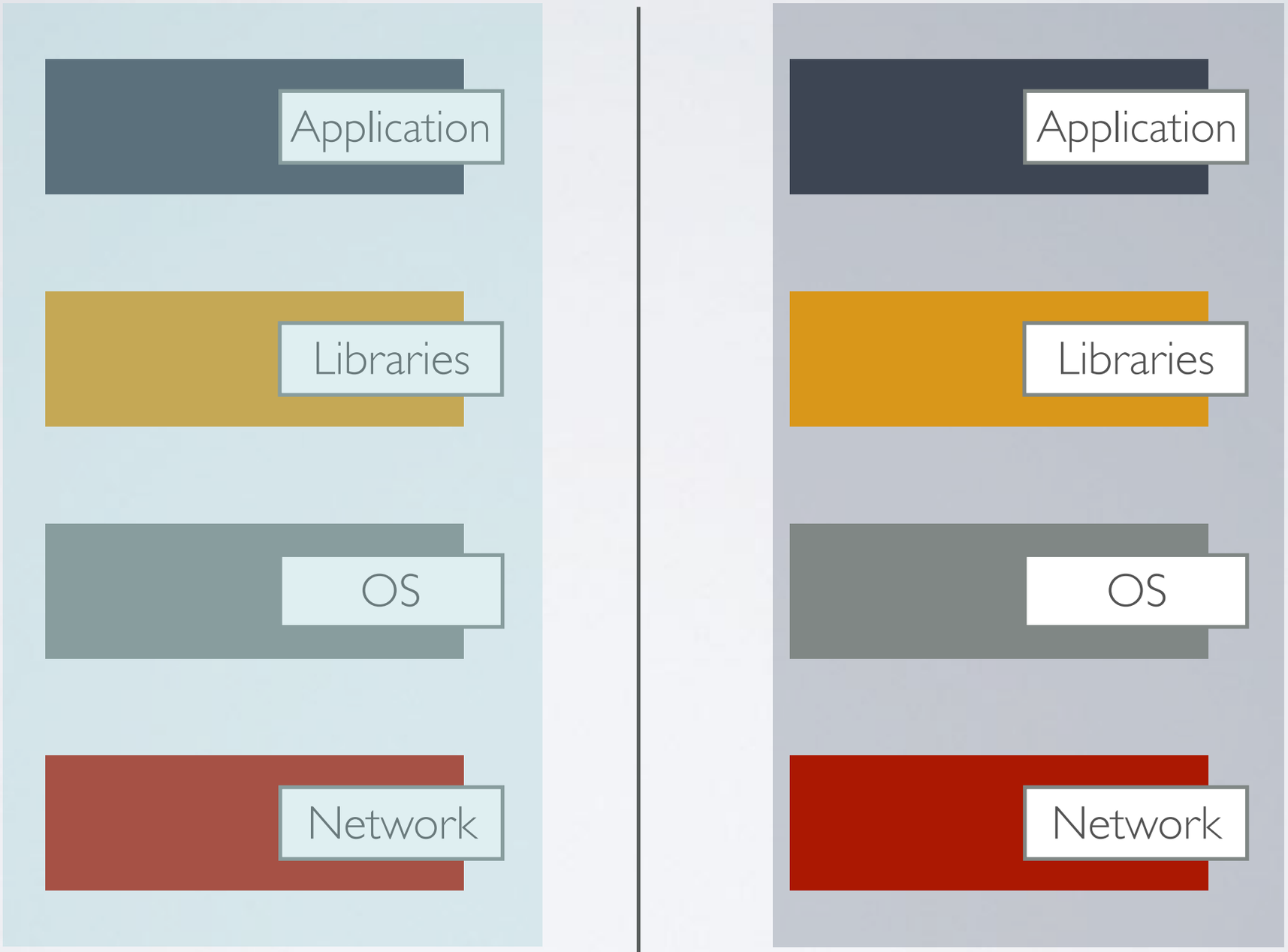
I HAVE NOTICED

- Tactical decisions are made in vertical teams
  - Sacrifice long-term value for short-term results
  - The consultant paradox
- Strategic decisions get made in horizontal teams
  - Build once, use everywhere
- Cloud experts are crazy expensive
- You cannot be really good at everything

BETWEEN DEPARTMENTS



BETWEEN APPLICATIONS



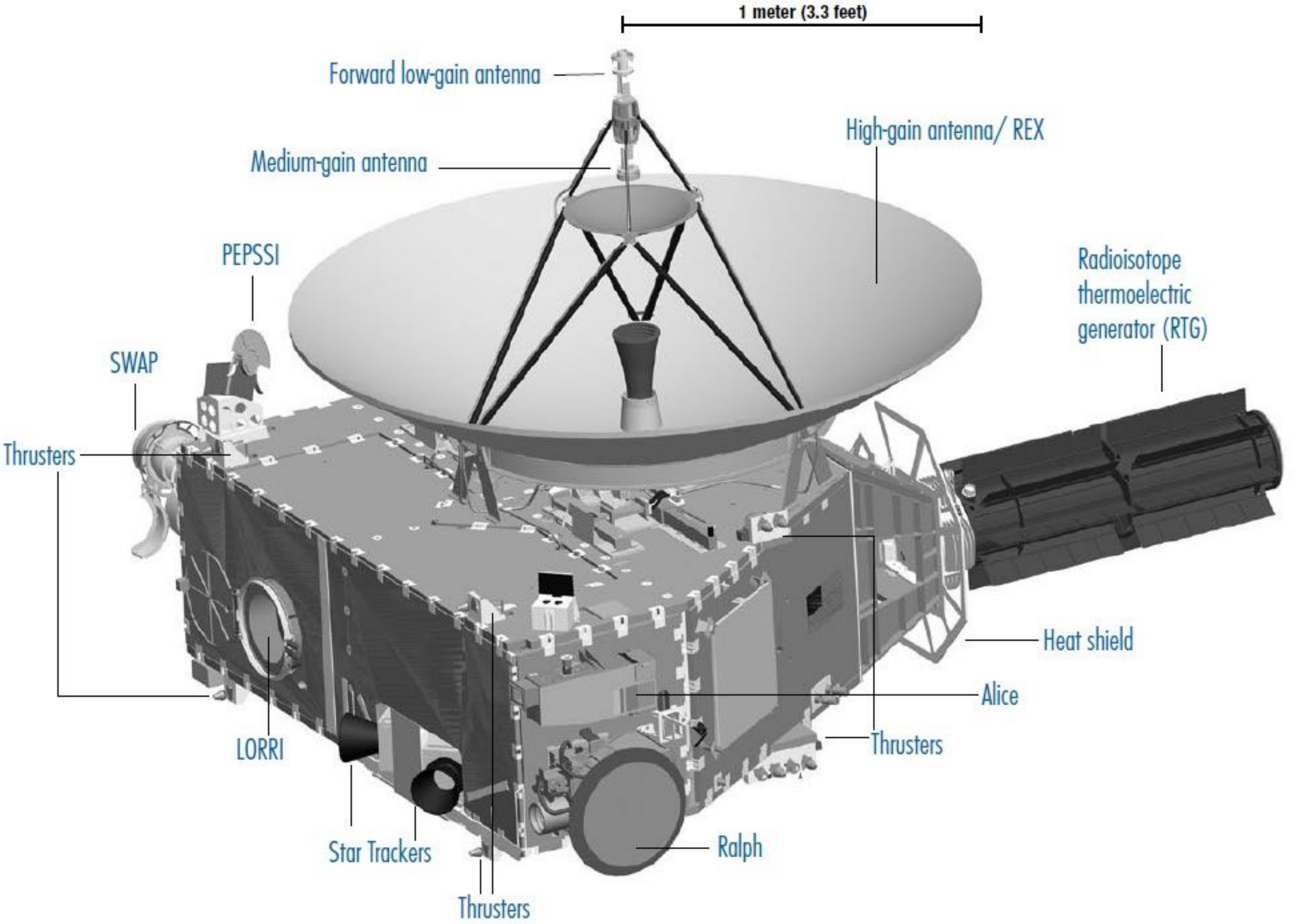
Application
Team 1

Application
Team 2

OK, Cliff… If you're so smart, what do you propose?

NASA?

They've been able to launch rockets… Maybe they have some ideas?


1 meter (3.3 feet)
Forward low-gain antenna
Medium-gain antenna
High-gain antenna/ REX
PEPSSI
SWAP
Radioisotope thermoelectric generator (RTG)
Thrusters
Heat shield
Alice
Thrusters
LORRI
Star Trackers
Ralph
Thrusters

Before building anything…

TECHNOLOGY PORTFOLIOS PREVENT MADNESS

- Cloud Formation
- OpsWorks (Chef)
- Lambda
- Git

TAKE TIME TO UNDERSTAND NETWORKING

- VPCs are like elaborate home networks
- RFC1918 networks everywhere
- NAT gateways
- Static routes
- If you do nothing else, govern your private space

"BASE OFFERINGS"

- Printing
- Authentication
- Users (POSIX)
- Groups (POSIX)
- Base Packages
- Logging
- NTP
- CIS Security Template
- Backups
- Housekeeping Scripts
- Patching
- Much More

- Can be anything form a full application to a single line in a configuration file

- Everything you need that is common across applications

- Usually expressed in CloudFormation and Chef

TREAT THEM LIKE SUPPORTED SOFTWARE PRODUCTS

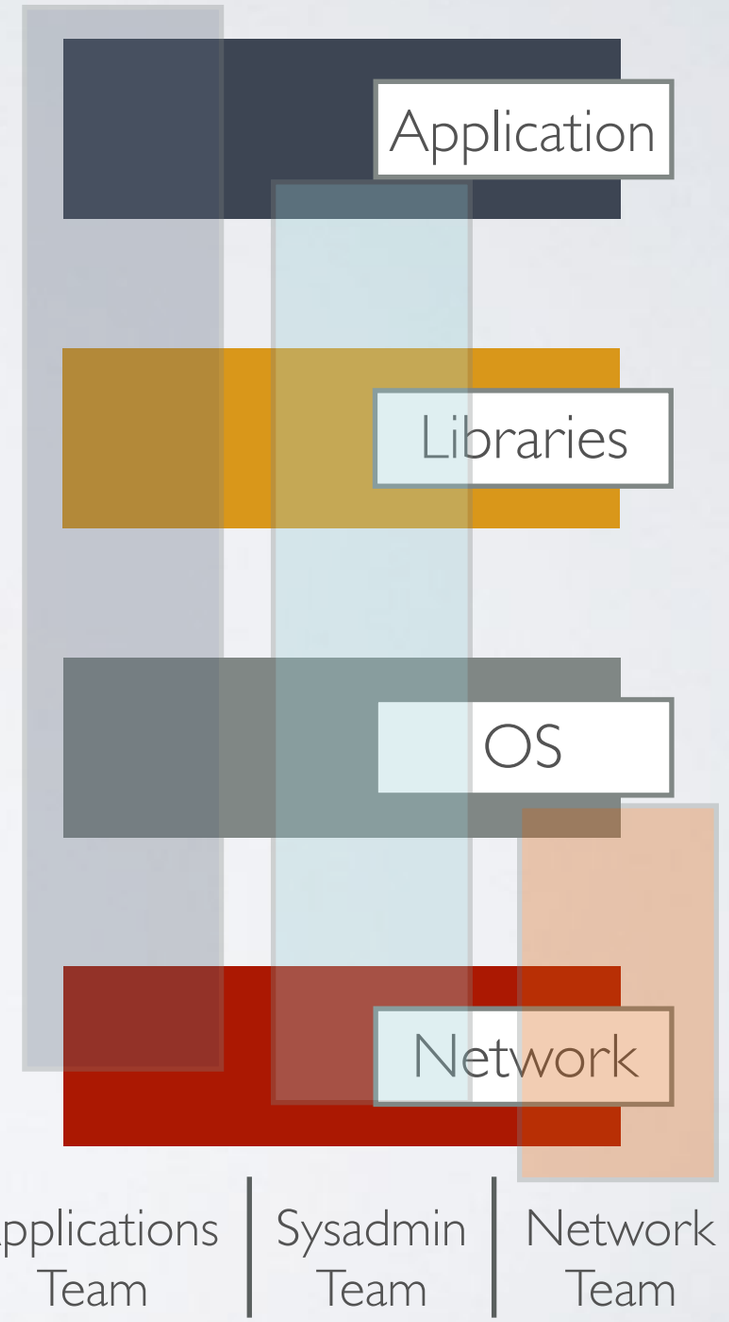
- Draw boxes around offerings & do them well
- Easy integration
- Accept and incorporate feedback
- Avoid introducing toxic changes
- Automate everything
- You don’t need to write them all

PLATFORM ENGINEERING PATTERN

- Offerings are written by distributed teams
- Owned by a service oriented team
- Application as a first-class citizen
- Offerings are treated as products

WHAT WOULD CONWAY PREDICT?

- Limited siloing around offerings
- Lots of communication flow between teams
- Reusable code
- Interchangeability between team members



SOUNDS GREAT, BUT…

- It is slower
- Experts can become fatigued
- People sometimes forget to ask before solving the same problem again
- The concept of "finished" is frustratingly vague
- Mistrust of the unfamiliar is real

MENTORSHIP

- Some team members will dive right in, others won't
- This is enough of a paradigm shift that strong engineers will feel intimidated (they are used to being good at things)
- Small offerings make for good training
- Screen sharing as a training tool

FINAL THOUGHTS

- Cloud providers promise turn-key AGILE (It's a lie)
- Two pizza teams will get you there, but are less helpful in operating the environment
- Platform engineers are closer to the "dev" than the "ops"
- The ability to move fast is something you earn
- Production is war, and war is hell (do your engineering up-front)
- Prioritize one-to-many relationships
- Make tools not widgets
- Have fun, build bridges and make friends on other teams

UC SANTA CRUZ cliffp@ucsc.edu

twitter @crpearson