

Acquia®

Effectively managing 1000 Drupal websites



#SolutionsArchitecture

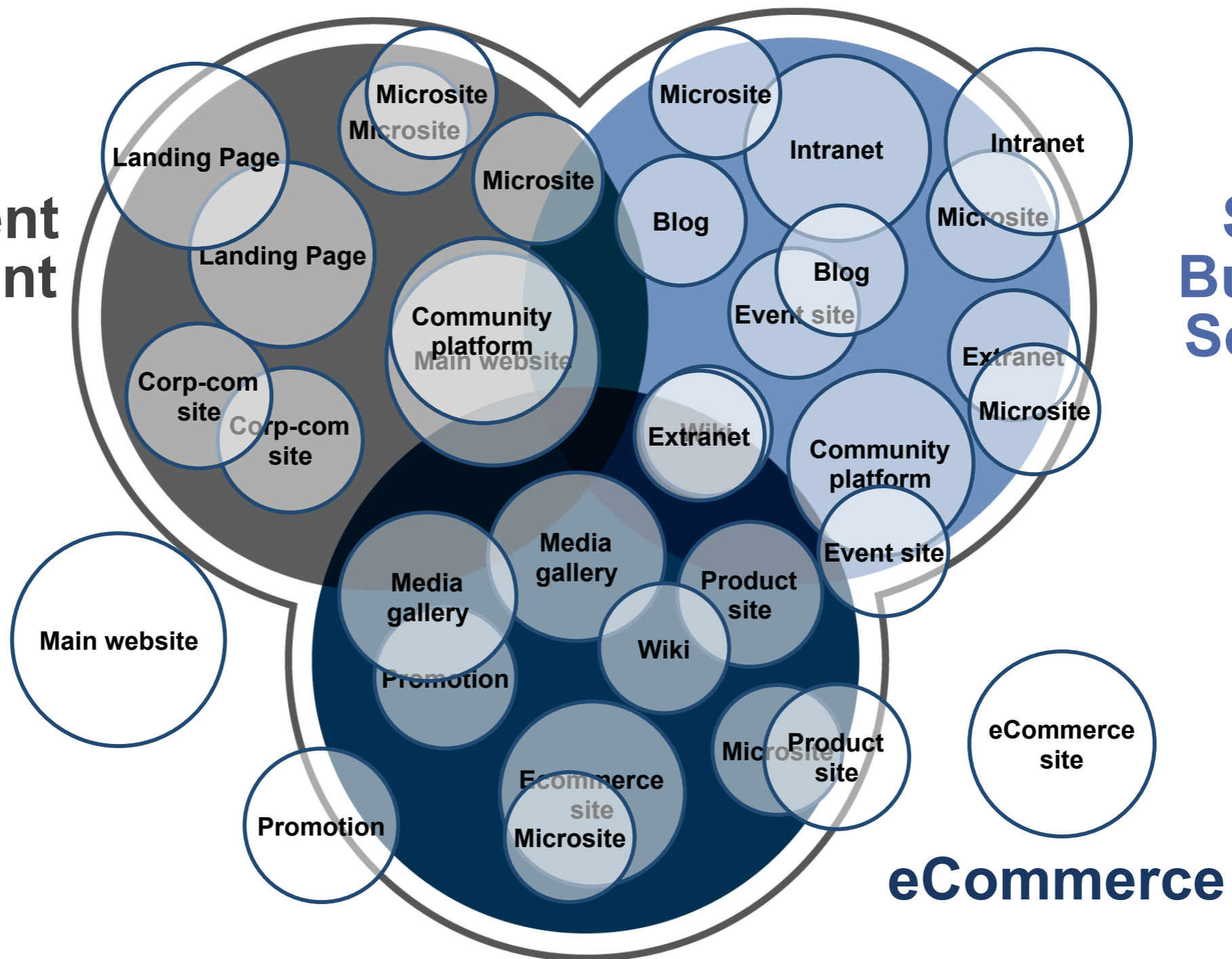
About me

- ▶ Solutions Architect @ Acquia
- ▶ Experience from:
 - ▶ Brightcove.com
 - ▶ Nokia MeeGo
 - ▶ Johnson & Johnson Open Solutions
- ▶ Twitter: @jakubsuchy



Web Content Management

Social Business Software





Acquia®

#SolutionsArchitecture

Monday, December 2, 13

4

Business requirements come from many teams, Resources are scarce, Team members come & go

You end up maintaining not one codebase but hundreds or thousands

This is the amount of time people typically spend on maintaining their site as opposed to creating new features. This is provided you do maintain your Drupal codebases :)

Now imagine you have 100 websites. You can easily imagine that without automation or some sane practices, you will end up having a full time person doing just site maintenance, That's not good & expensive



5-20%

Acquia®

#SolutionsArchitecture

Monday, December 2, 13

4

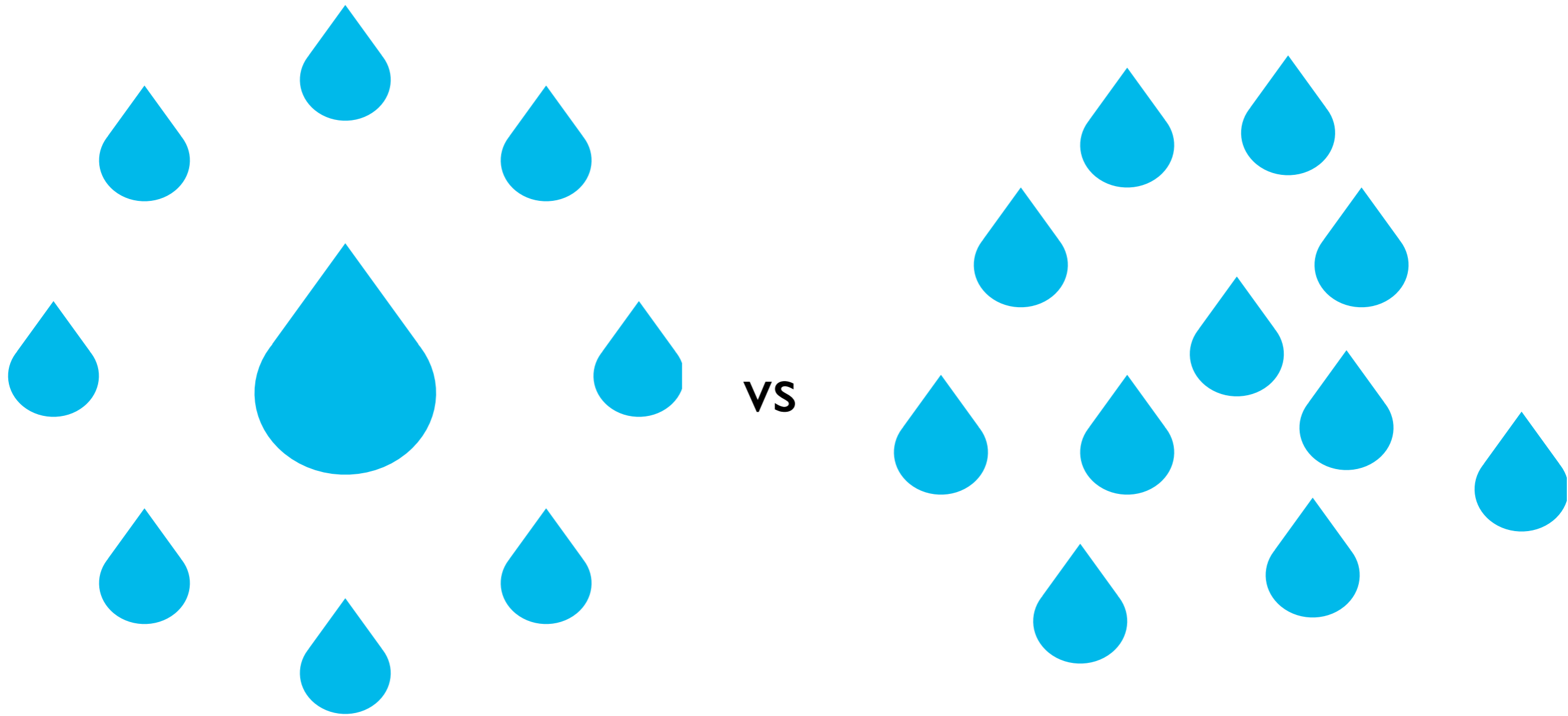
Business requirements come from many teams, Resources are scarce, Team members come & go

You end up maintaining not one codebase but hundreds or thousands

This is the amount of time people typically spend on maintaining their site as opposed to creating new features. This is provided you do maintain your Drupal codebases :)

Now imagine you have 100 websites. You can easily imagine that without automation or some sane practices, you will end up having a full time person doing just site maintenance, That's not good & expensive





- they typically have either a centralized IT team that maintains and owns all sites - can influence, dictate
- decentralized - many different teams and different responsibilities, each owning their own sites

Imagine a 7 figure sum

Our goals

- ▶ Reduce site maintenance
- ▶ Increase efficiency
- ▶ Foster collaboration

Establish a governance model

- ▶ Single team as a gatekeeper
- ▶ The trick: Balance of power for site teams
- ▶ Only works for centralized teams

Why do you use Drupal?



Acquia®

#SolutionsArchitecture

Monday, December 2, 13

11

Why do you use Drupal? For the community!

This is the best scenario especially for decentralized teams

Consider creating an internal open source community. Running your own open source community can foster collaboration, enable your teams to reduce maintenance overhead and increase speed of development

#9 BREAK DOWN BARRIERS



Create an internal community

- ▶ NBC Universal
- ▶ University of California, San Francisco

Join LSD - largescaledrupal.com



LARGE SCALE DRUPAL

Acquia®

#SolutionsArchitecture

Take baby steps

- ▶ Organize a Camp! Hello!
- ▶ Schedule office hours
- ▶ Github all the things
- ▶ Let a leader emerge

Site (Yellow):

/sites/foo/modules/*
/sites/foo/themes/*
/sites/bar/modules/*
/sites/bar/themes/*

Platform (Brown):

/profiles/foo/modules/*
/profiles/foo/themes/*

Core (Blue):

/modules/*
/themes/*
/includes
... etc.

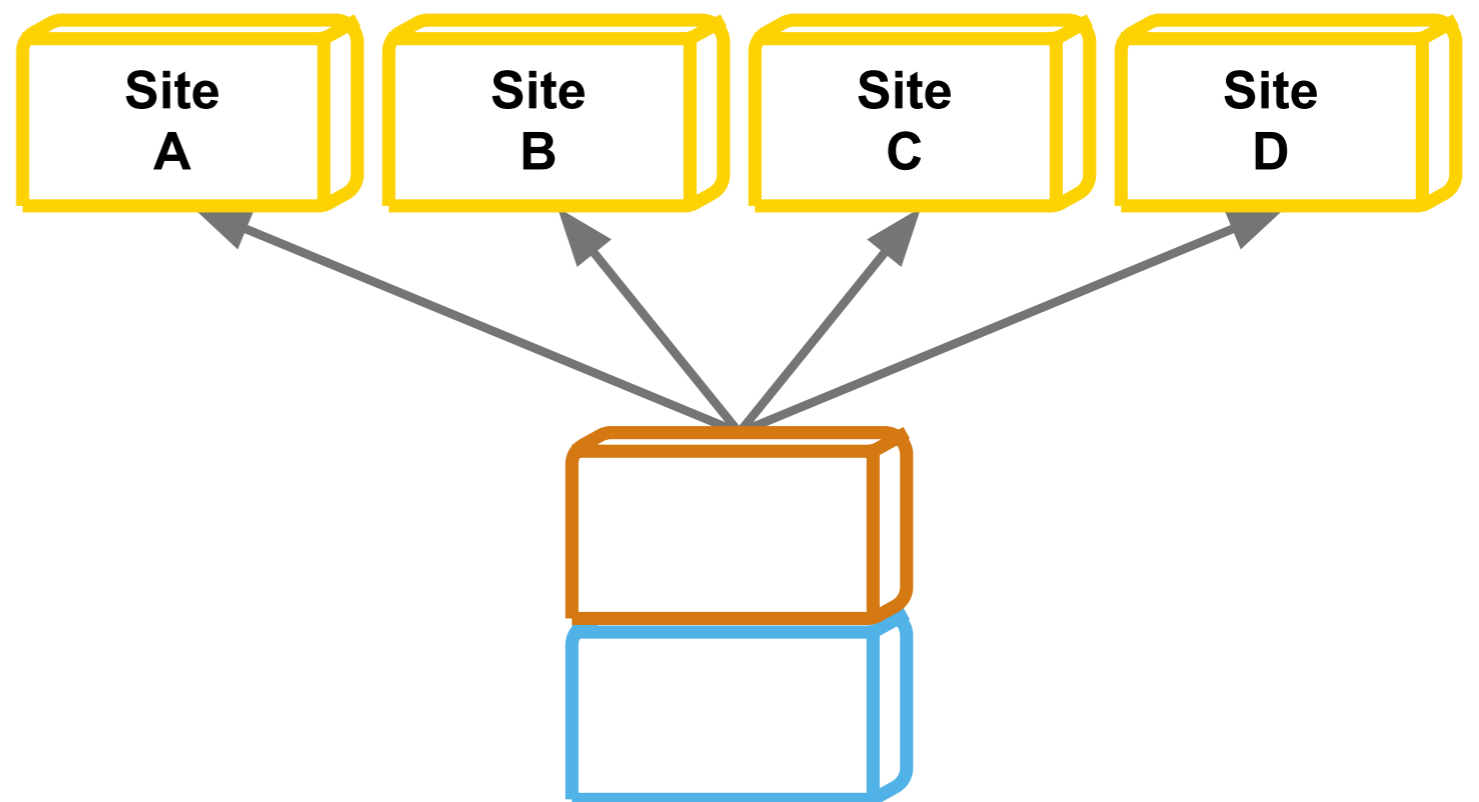
Site specific themes & modules

Platform theme & modules

Drupal core

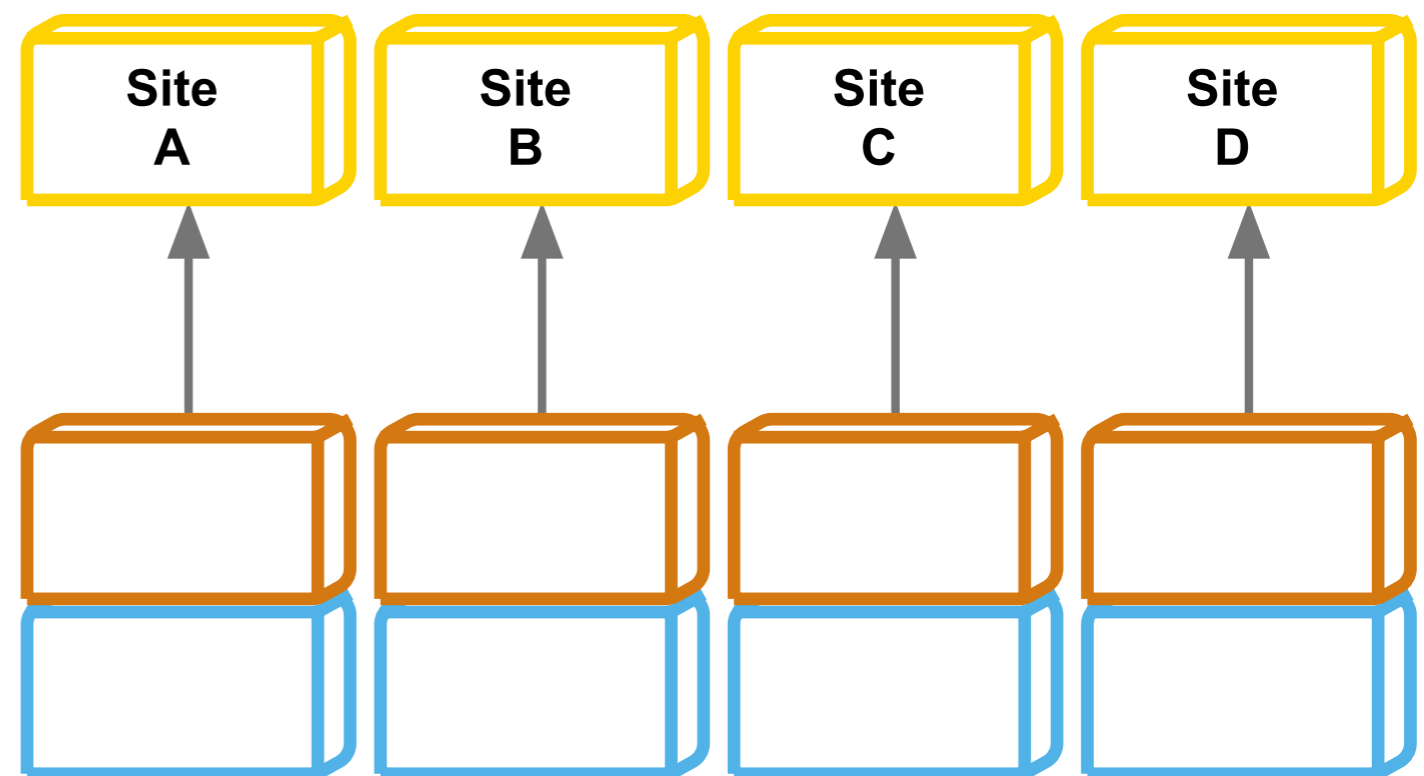
Multi-Site

- ▶ Works great for single teams
- ▶ Easy to keep core & platform in sync
- ▶ Hard to test & QA across many sites
- ▶ Difficult for many teams



Independent sites

- ▶ Lets each site be tested, QA'd individually at their own pace
- ▶ Lets independent teams work on individual sites without breaking others
- ▶ Difficult to maintain



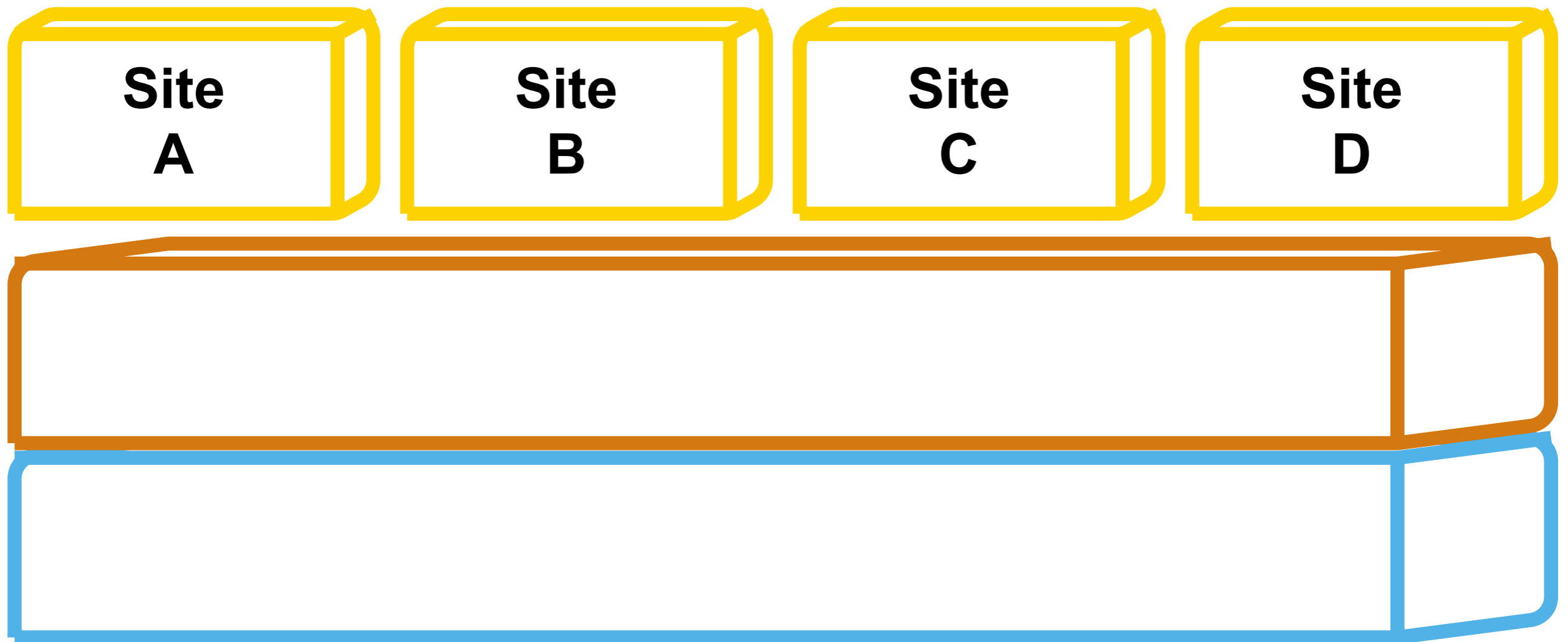


Acquia®

#SolutionsArchitecture

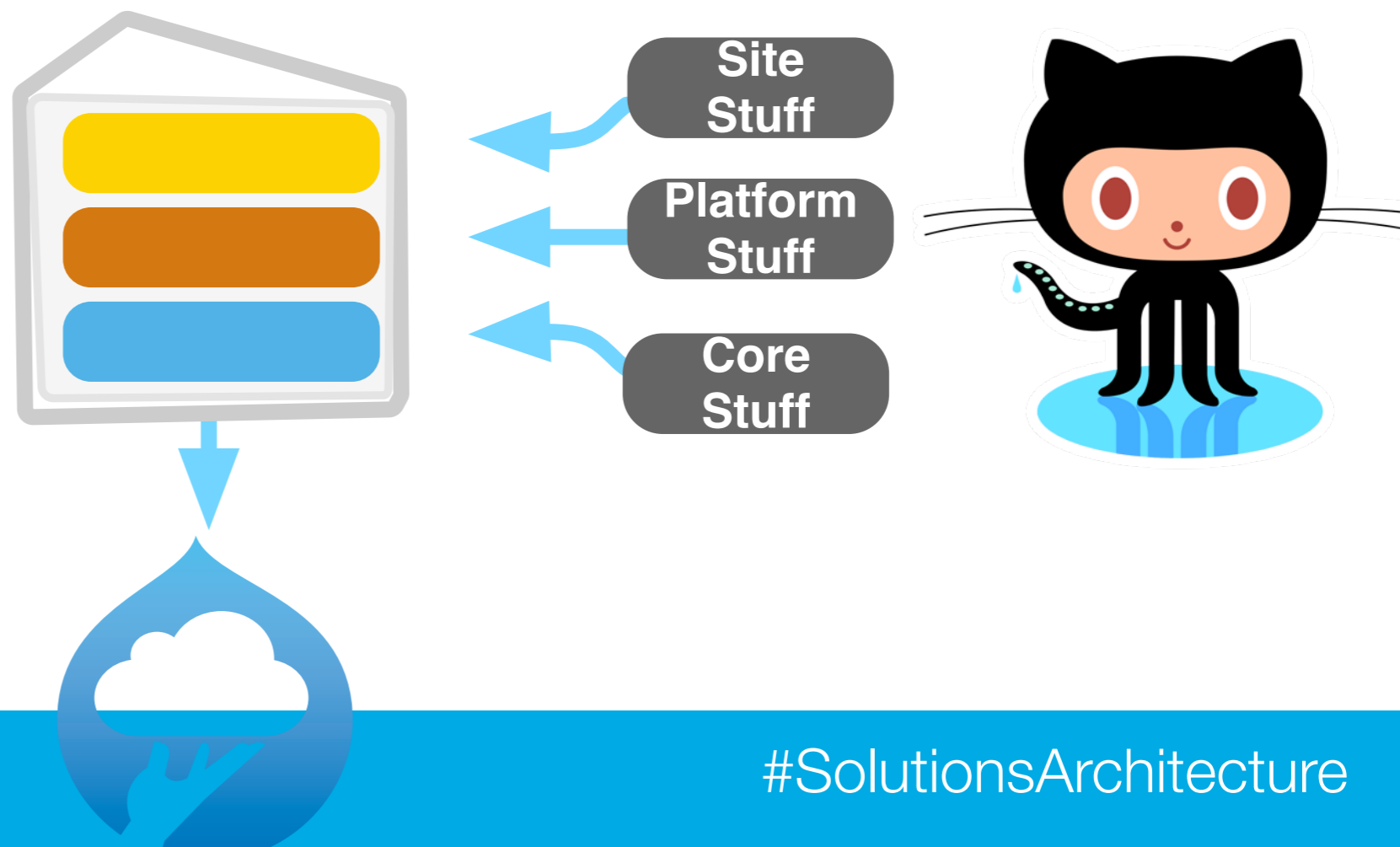
Platform / Distribution approach

- ▶ Marry multi-site with independent codebases
- ▶ (on code level)

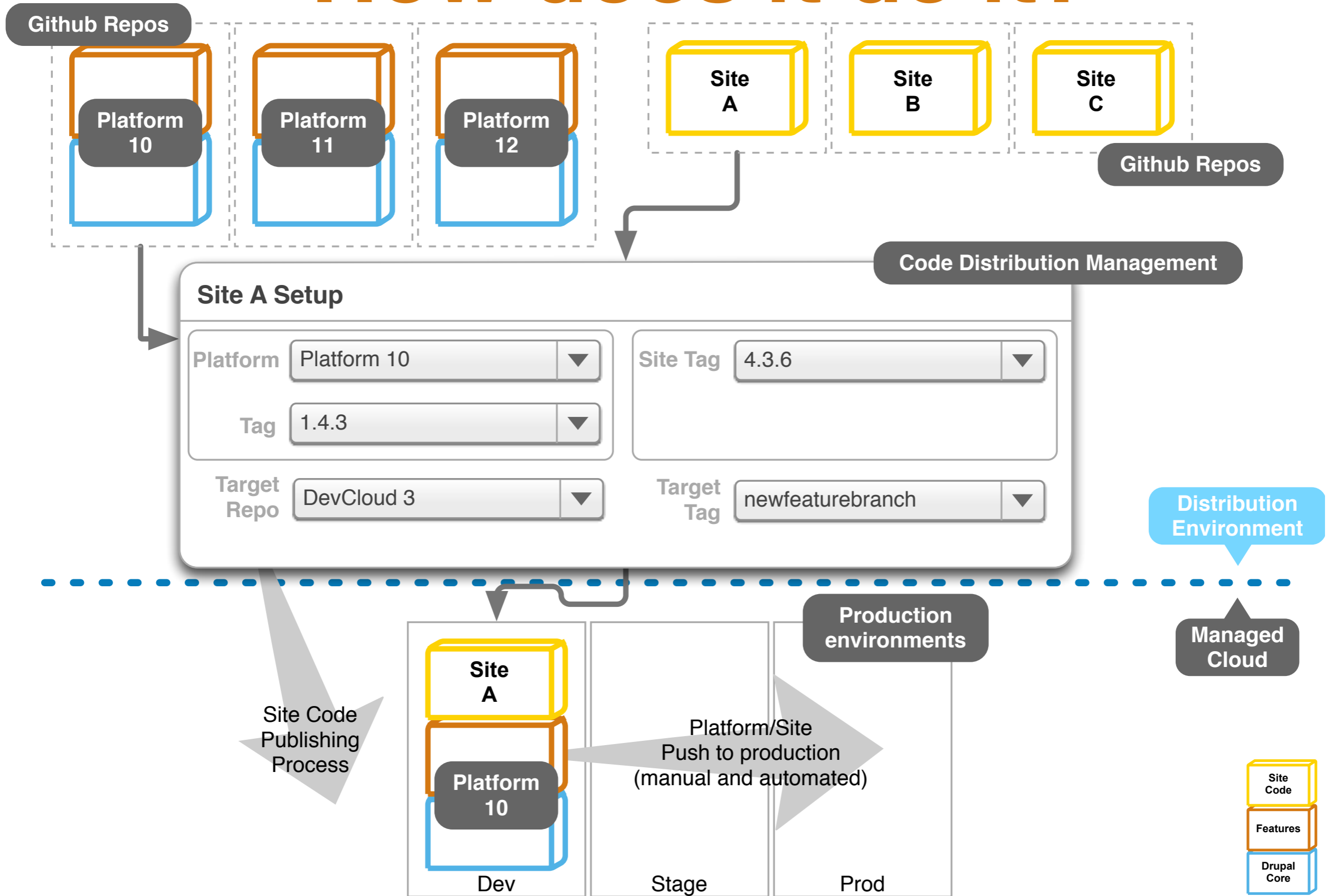


Mmm...Cake

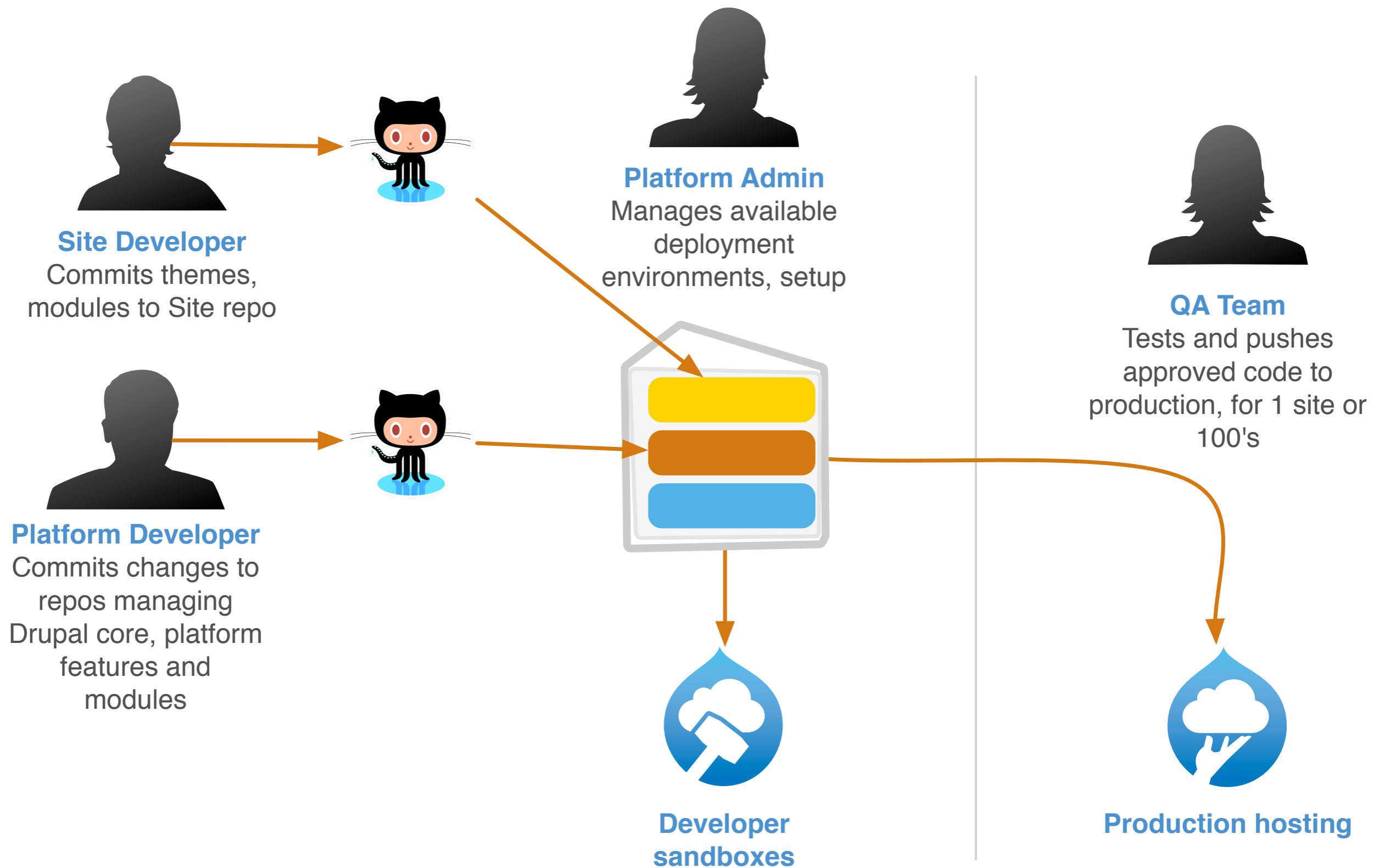
- ▶ Segregate responsibility for site-specific development, platform development and QA
- ▶ Let dev teams use the tools they're used to (git, Github, Drupal)



How does it do it?



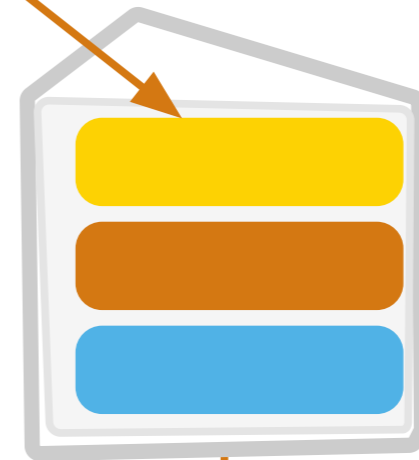
Who does it work for?



Developers!



Developers
Commit to repos
they've been granted
access to in Github



Use Layer Cake to
deploy their repos
periodically to
sandboxes



Developer's personal
sandbox where they can test
against replica of production
stack

Admins!

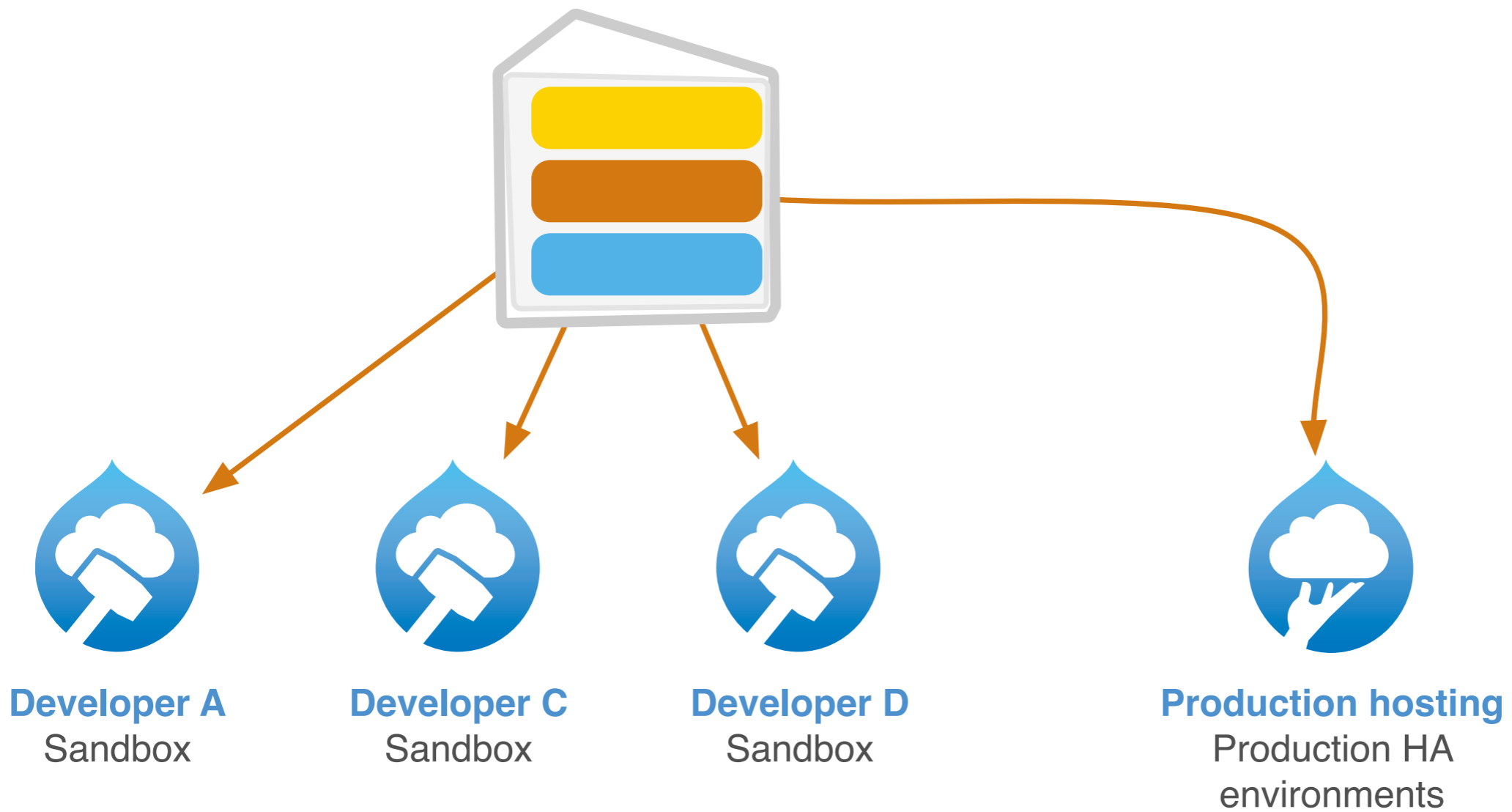


Platform Admin

Sets holds the keys to creating new mappings

Site A Setup

| | | | |
|-------------|-------------|------------|------------------|
| Platform | Platform 10 | Site Tag | 4.3.6 |
| Tag | 1.4.3 | | |
| Target Repo | DevCloud 3 | Target Tag | newfeaturebranch |



Quality Assurance!



QA Team

Tests and pushes approved code to production, for 1 site or 100's



Production environments

Filter Sites

Keyword

Platform

Deploy Stage -> Prod

| Site | Platform Version | Site Version |
|--|------------------|--------------|
| <input checked="" type="checkbox"/> Site A | 10b | 4.5 |
| <input type="checkbox"/> Site B | 10b | 3.1 |
| <input checked="" type="checkbox"/> Site C | 10b | 1.0 |
| <input checked="" type="checkbox"/> Site D | 10b | 7.2 |
| <input checked="" type="checkbox"/> Site E | 10b | 134.2 |

Deploy

Cancel

Site management gotchas

- ▶ Testing the upgrade path on all sites. Shoot for eventual consistency but beware of the dangers of falling out of sync.
- ▶ When using the Layer Cake approach beware Platform and Site compatibility issues.

How to start

- ▶ Do an inventory
- ▶ Start just by moving modules around.
- ▶ Standardize on top modules.
- ▶ Once modules are standardized, start capturing your common configurations into Features.

How many sites do we have?

How similar are they?

Are we looking at one platform or many?

Features. Love them. Hate them. Use them.

REVERT ALL THE



Acquia®

#SolutionsArchitecture

Overriden Features

- ▶ Problem: Feature overridden due to configuration changes done in dev, stage or prod.
- ▶ Fix: You're doing it wrong!
- ▶ Adopt a proper dev-> stage -> prod workflow.
- ▶ Automatically revert Features on deployments. Script drush fr.
- ▶ “drush fu” your workflow. (best Drush command ever!)

Aaaand...

► It's all about to change in Drupal 8

[Home](#) » [Administration](#) » [Configuration](#) » [Development](#)

Synchronize configuration

Import configuration that is placed in your staging directory. All changes, deletions, renames, and additions are listed below.

1 changed

| Name | Operations |
|---------------------|----------------------------------|
| views.view.articles | View differences |

[Import all](#)

SaaS: Acquia Site Factory

1 Select website template



- Information architecture
- Page layouts
- Roles & permissions
- Content types & display
- Pre-defined themes
- 3rd party integrations

2 Configure site-specific settings



3 Add design & site content



Done.



[http://www.\[customdomain\].com](http://www.[customdomain].com)

What about if you stop maintaining code whatsoever? There are platforms such as Acquia's Site Factory that let you create hundreds of sites without any code if your sites are easy enough.

What have we learned

- ▶ Managing a lot of sites is hard. Deal with it, budget for it
- ▶ Foster an internal community
- ▶ Create a platform / distribution
- ▶ Featurize your code
- ▶ Consider SaaS

Acquia®

Thank you, questions?

Jakub Suchy, jakub.suchy@acquia.com
@jakubsuchy



#SolutionsArchitecture