



SRE UIs

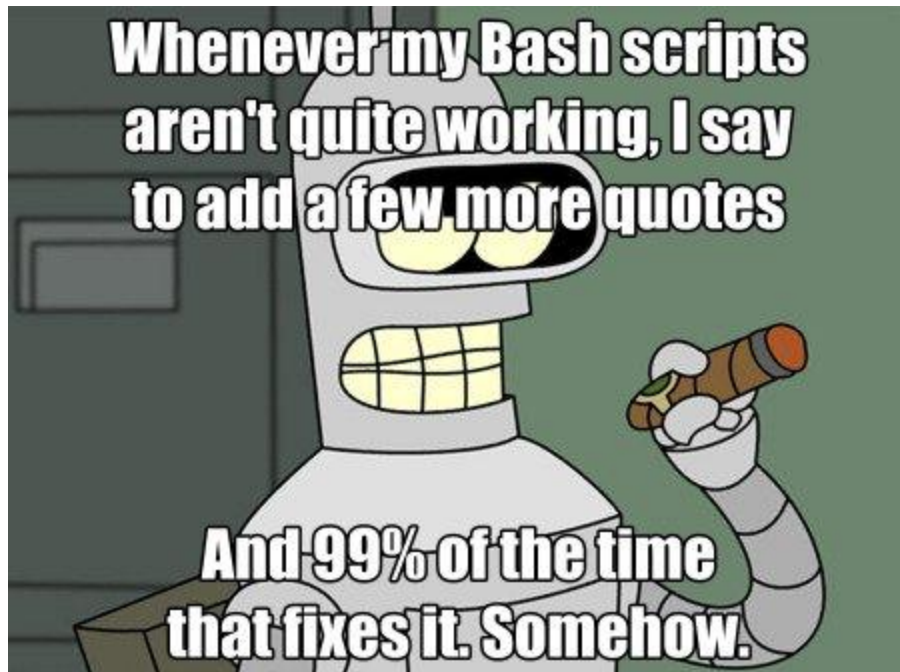
Transcending the CLI
Michael Avrukin - Google

SRE designing UI:

v

Agenda

- Get your hands off my CLI!!!
- Who are you users?
- GUIs are really not that hard...
- Architect with user in-mind from the start
- Build a toolbox of components
- Profit!



So what's really wrong with the CLI?

First question - who is the user?

Well... I'm the user!

A Story of a successful tool

- I once needed to find something in a log file
- So I used `grep`
- Then I needed to find something in lots of log files
- So I used `grep -r`
- Then I needed to find something in lots of log files for service x
- So I used `find . -name "*x.log" -exec grep`
- Then I wanted to find something in lots of log files on lots of machines
- So I started a company, and called it Splunk
- PROFIT

splunk> App: Search & Reporting Nate McKervey Messages Settings Activity Help Find

Search Pivot Reports Alerts Dashboards Search & Reporting

New Search

Save As Close

sourcetype=access* status>=400 Last 60 minutes

✓ 2,124 events (1/14/15 10:48:00.000 PM to 1/14/15 11:48:58.000 PM) Job Smart Mode

Events (627) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect 1 minute per column

180 140 100 60 180 140 100 60

10:50 PM Wed Jan 14 2015 11:00 PM 11:10 PM 11:20 PM 12 minutes 11:42 PM

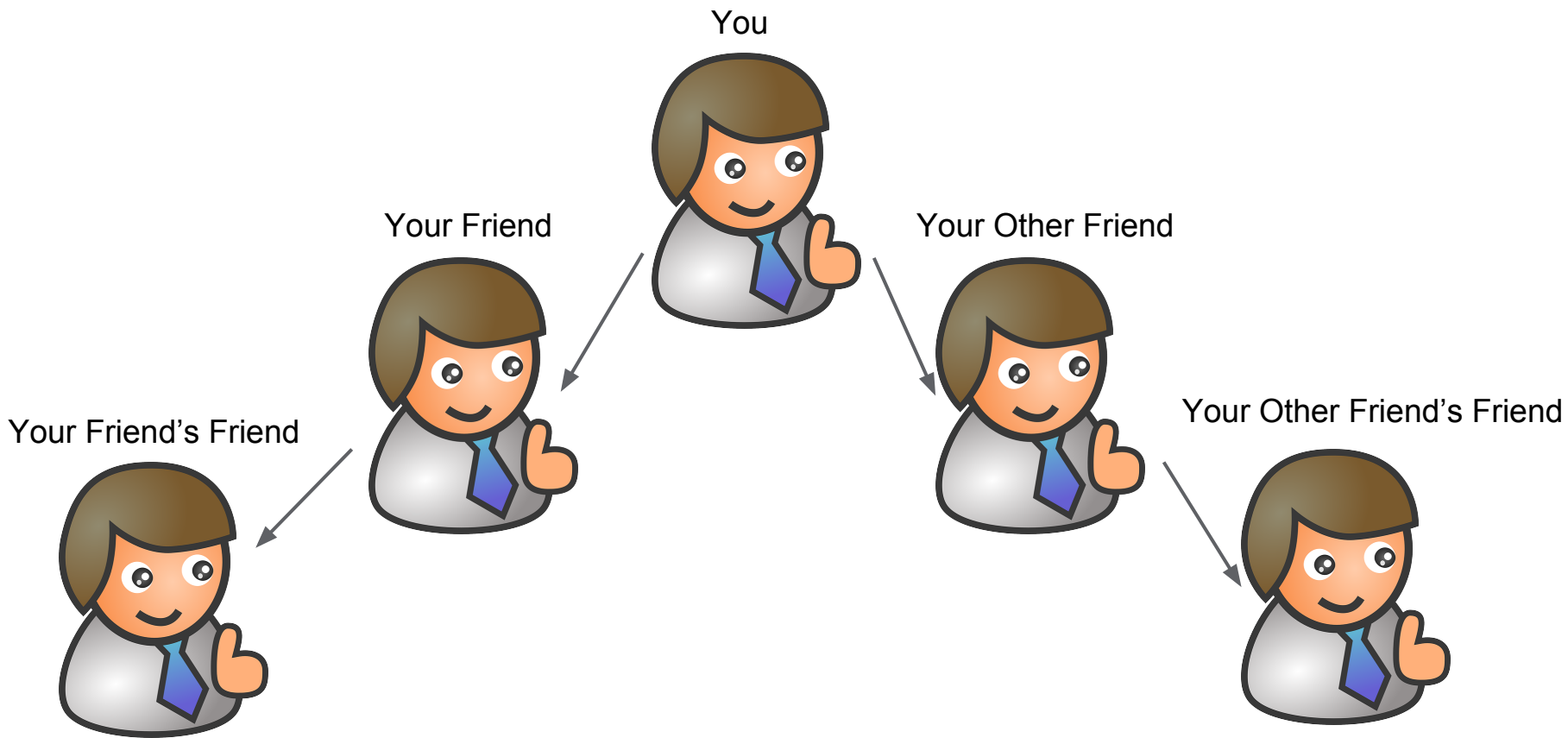
List Format 20 Per Page < Prev 1 2 3 4 5 6 7 8 9 ... Next >

< Hide Fields All Fields

Selected Fields
 a host 2
 a source 2
 a sourcetype 2

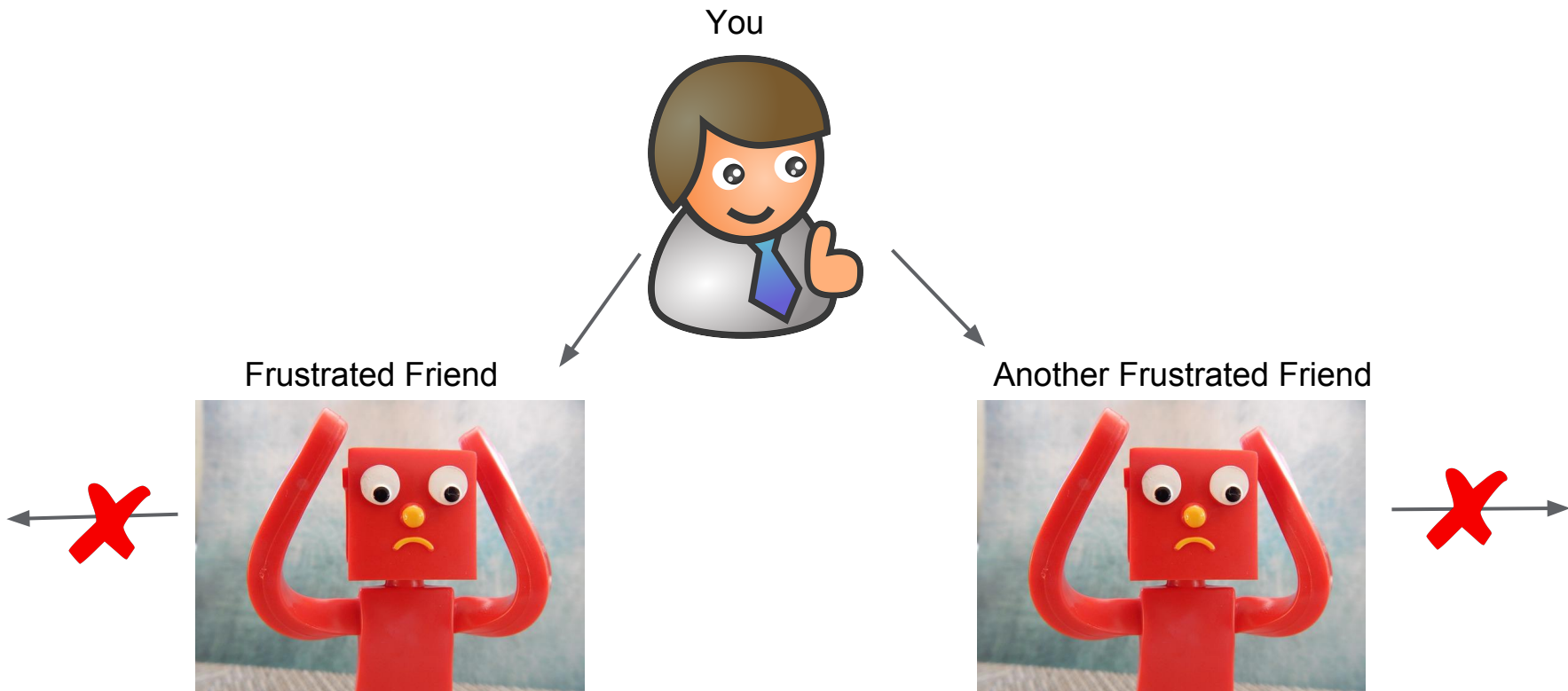
i	Time	Event
>	1/14/15 11:41:58.438 PM	10.2.1.34 27.160.0.0 - - [14/Jan/2015 23:41:58:438509] "GET /oldlink?item_id=EST-15&JSESSIONID=SD3SL5FF8ADFF3 HTTP 1.1" 404 333 "http://shop.splunktel.com/category.screen?category_id=BATTERIES" "Mozilla/5.0 (iPhone; CPU iPhone OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3" 1718 host = dmzlog.splunktel.com source = /opt/apache/log/access_combined.log sourcetype = access_combined

The Step Before Starting A Company...



Your Tool Is Beloved!

But what if your tool works as intended... but has a bad interface?



So you say - “CLIs are much less frustrating than bad UIs”

Well... that is true sometimes, but most of the time, a very small group of people actually know how to operate it.




CLI Limitations

- Interface is not clear
- Is it `-help` or `--help` or `--?` or `-?` or `help` or `<blank>`
- What about “non technical users”? They do exist! They could be your manager, or your manager’s manager.
- What is the input format?
- What is the output format?
- Where is the progress indication?
- How do I know when flags are mutually exclusive?

You can build bad UIs as well, but the issues are more glaring and there are less places where to hide

UIs are not that hard, I promise

State of the world

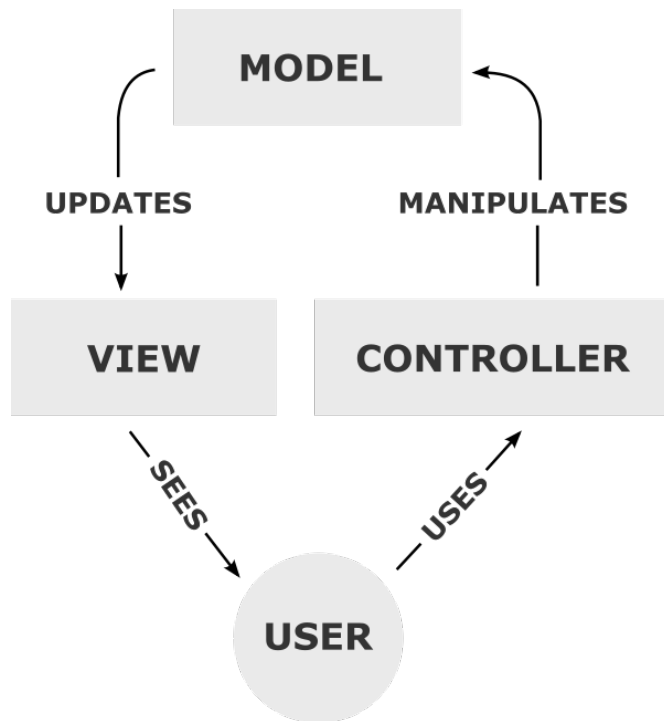
- Focus on Web UIs, they also work on mobile if done right
- Prototype & Connect Pieces Quickly:  ANGULARJS
by Google
- Make it look pretty - easily: 
- Keep your backend
- Template UI - Closure Templates aka Soy: 

Basic UI Concepts

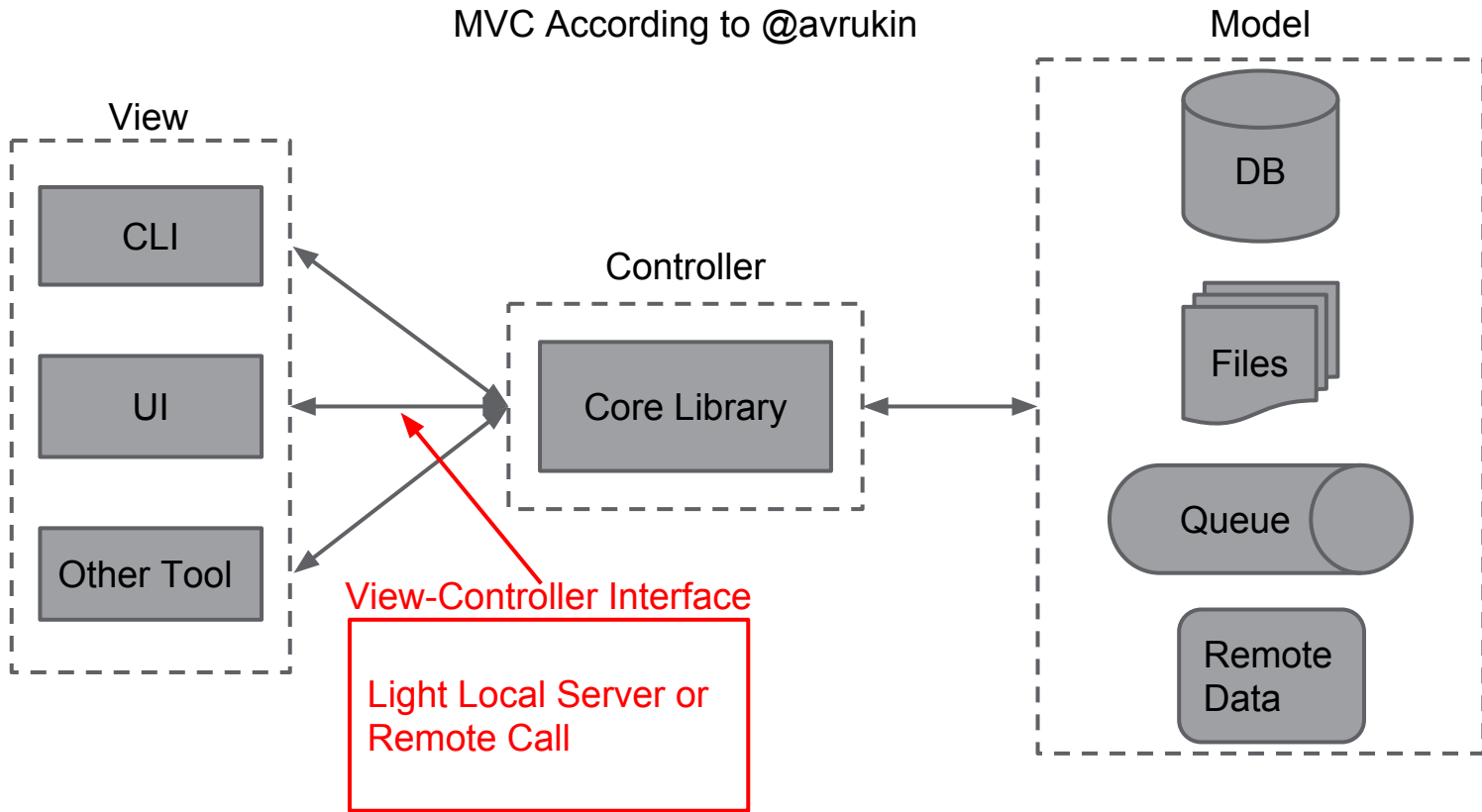
- Minimize Clutter - Break information into palatable pieces
- Flow - Top-Down & Left-to-Right (or Right-to-Left)
- Specific set of choices in drop-downs or radio button
- Free text input should be minimized
- Highlight important areas
- Minimize Text - Easy to read, large font
- Give your UI to a neighbor to test drive

What really matters is architecture

MVC According to Wikipedia



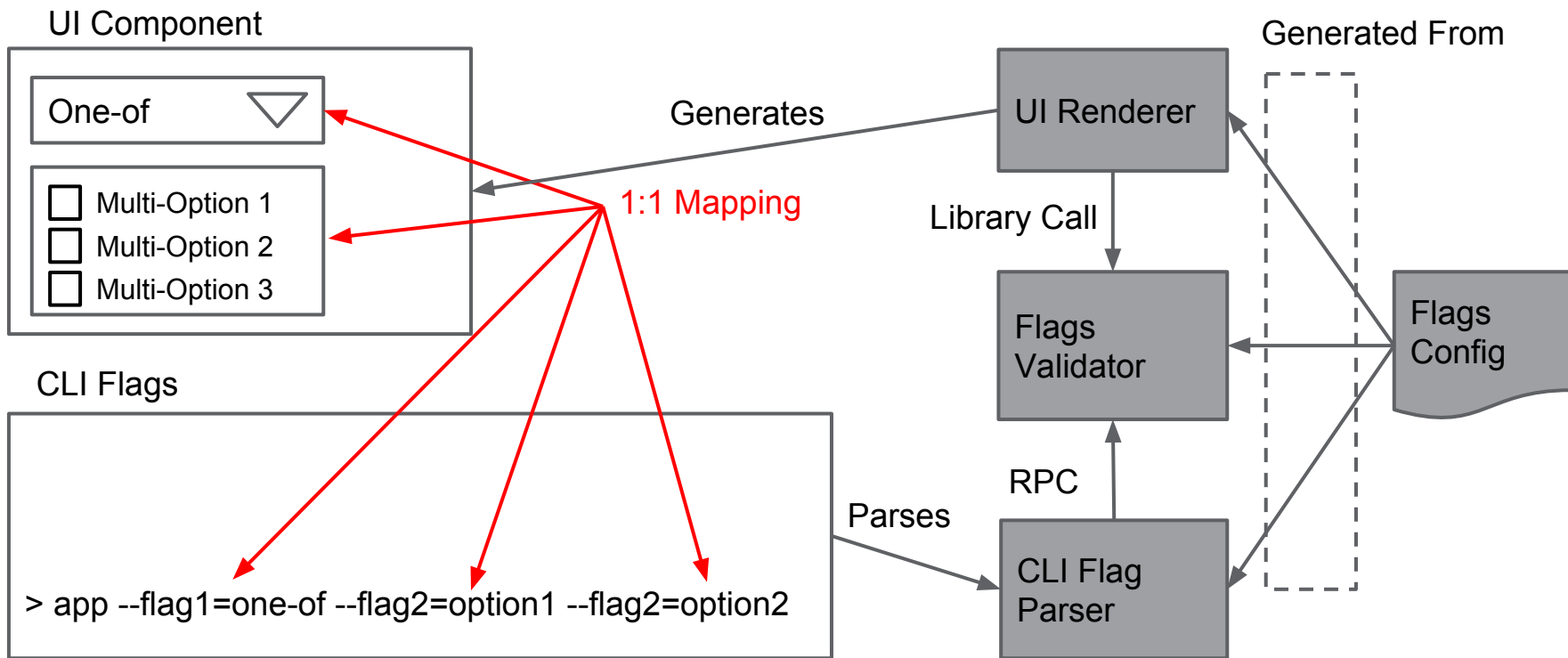
MVC According to @avrukin



Your Toolbox

You can build a small set of reusable patterns and tools,
don't have to start from scratch every time.

Example: UI Component + Command Line Flag



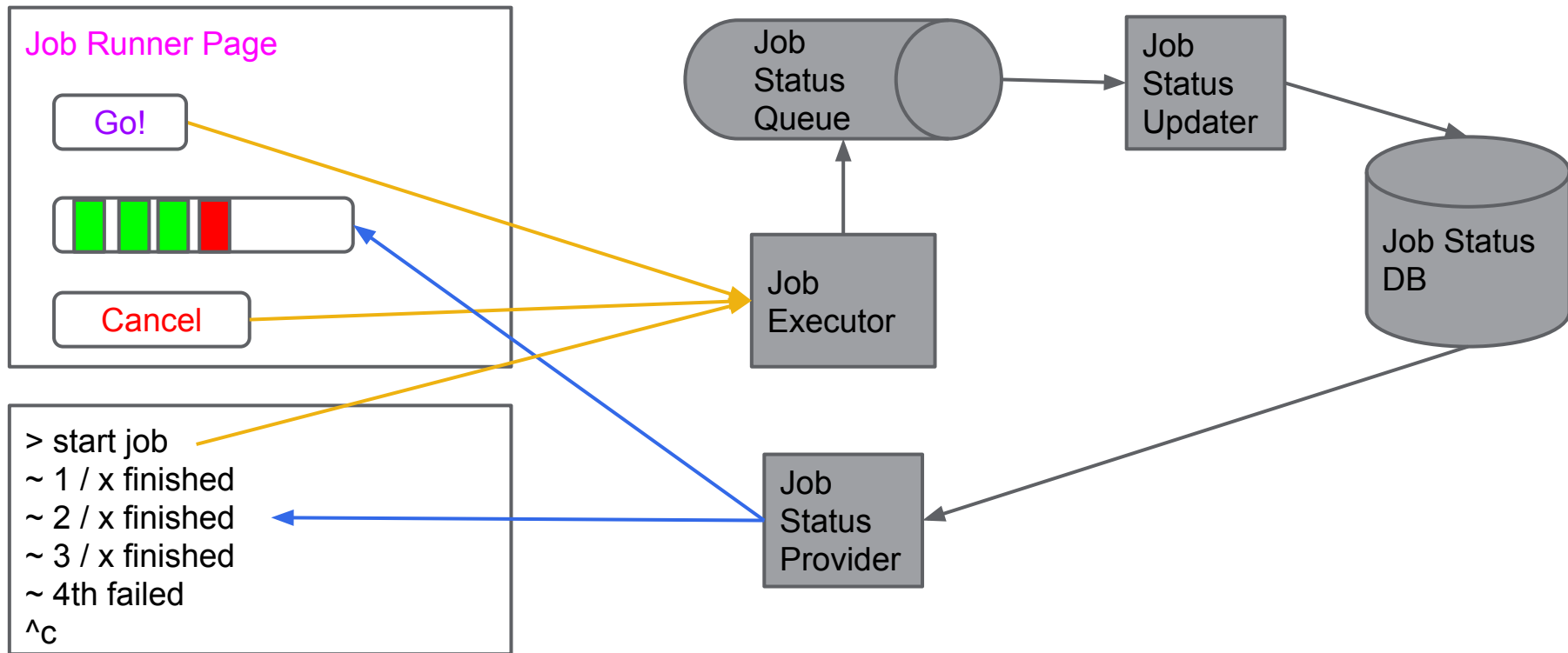
Example: Deploying a binary to production across many servers

Progress Provider and Monitors - Deploy Image to Servers

- A built image is placed in storage - S3
- Trigger of image deployment
 - Note the number of machines to deploy to
 - For each deployed machine, publish status to shared db - (Progress Provider)
- CLI tool can monitor progress through RPC (Progress Monitor)
- UI can monitor progress through JSON API (Progress Monitor)
- Additional actions - Cancel & Rollback
 - CLI - ctrl-c
 - CLI - another console
 - UI - Big red button

Architecture is Generic

Progress Provider & Monitor Architecture



Some Common Components - Use Architectural Patterns

- Progress indicator: Progress Provider + Progress Broadcaster
- File Parser
 - Configuration Parser + Viewer + Validator
 - Log Parser + Indexer + Searcher + Viewer
- Remote State: Process Viewer + System State + *Providers
- Dependency Resolution + Interconnection
- UI
 - Tabs
 - Input Configuration - Dropdowns, Radio, Checkbox
 - Fields - Typed - with validation

Thank You for Listening

Q & A