CS130: Software Engineering

Lecture 17: Deployments, Experiments, and Launches

Central topic for today's lecture: New Functionality

Let's say your product has launched and is serving live traffic.

- How do you add new functionality without causing downtime for existing users?
- How do you know if new functionality was the right thing for your users?

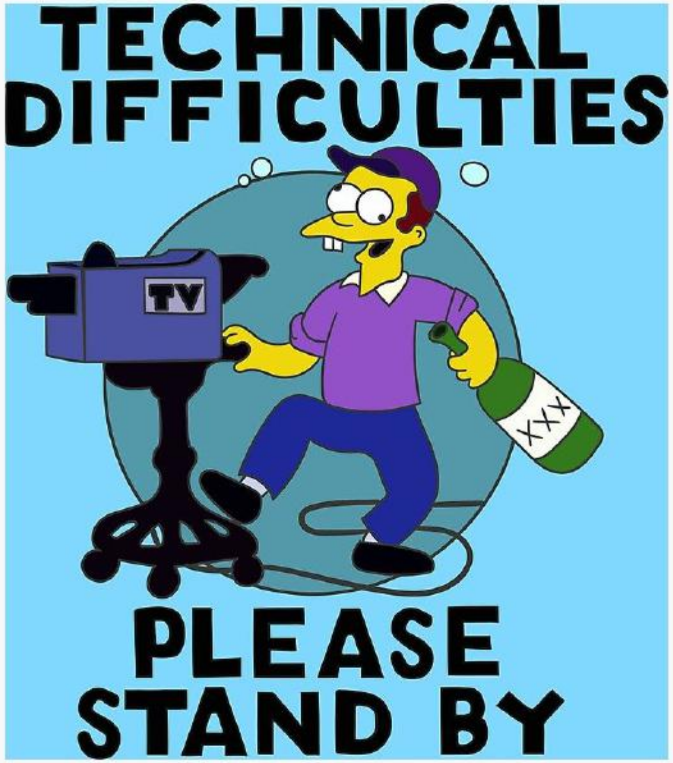
celonis

https://www.celonis.com/careers/jobs/

Deployments

Safety in numbers

- We've talked in depth about deployments
- How many of you have deployed your webserver at some point this quarter in a broken state?



Safety in numbers

- We've talked in depth about deployments
- How many of you have deployed your webserver at some point this quarter in a broken state?
- **What if your webserver was serving traffic when you deployed a broken version?**

Safety in numbers

- Let’s have more than one deployment of our server!
- Common deployment types:
  - Devel
  - Staging
  - Canary
  - Prod

Prod

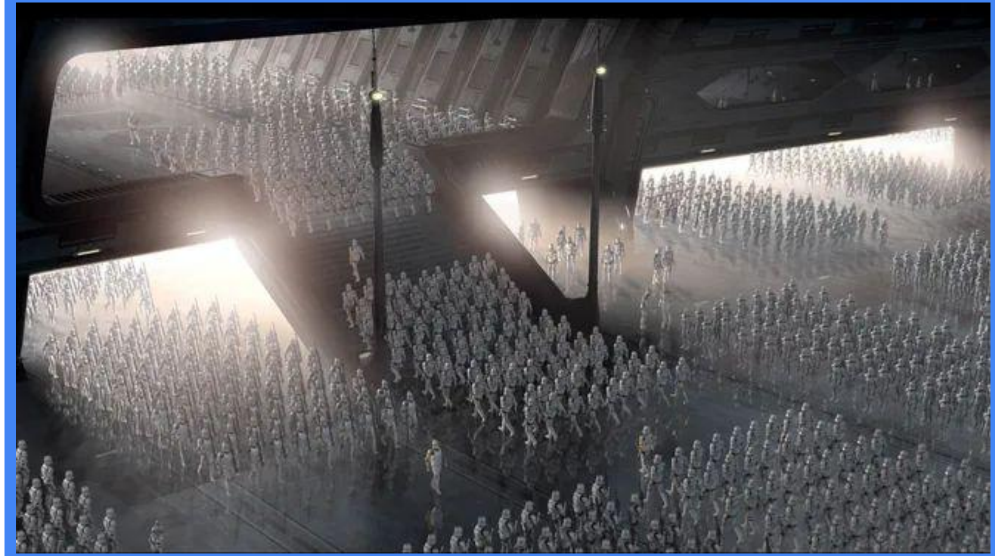
- Short for “production”
- All user traffic goes here
- Outages or downtime here leads to huge costs!

Canary

- "Canary in the coal mine"
- Small amount of user traffic goes here
- Outages or downtime here are less costly
  - Don't need to fix canary
  - Divert traffic to prod
  - Doesn't need to be up all the time

Staging

- As in “staging area”
- Deploy here what you intend to deploy to production
- Useful for QA testing to gain confidence that your deployment will work
- Connects to production backends where possible



Develop

- Has the latest version of your code
    - E.g. Continuous deployment from HEAD, deploy once a day from HEAD
- Used by developers to test new functionality in a deployed setting
- Usually connects to devel backends

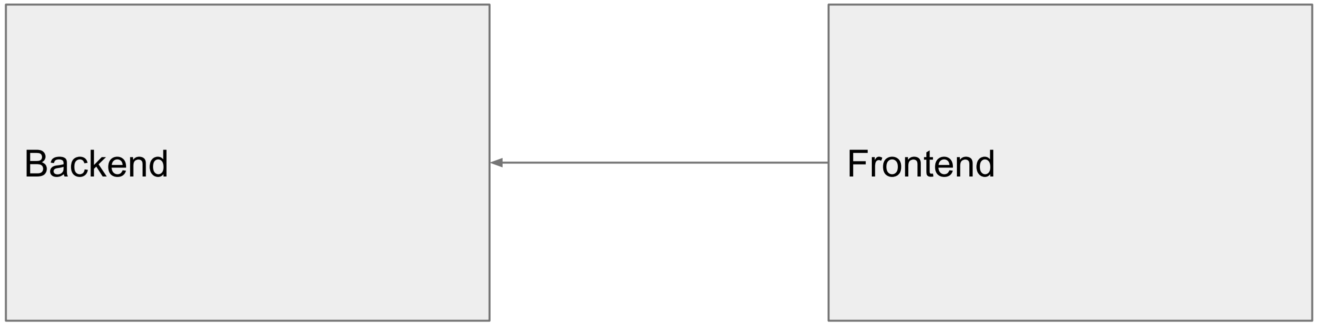
Version (in)compatibility

“No downtime”

- Your system will likely involve multiple different binaries

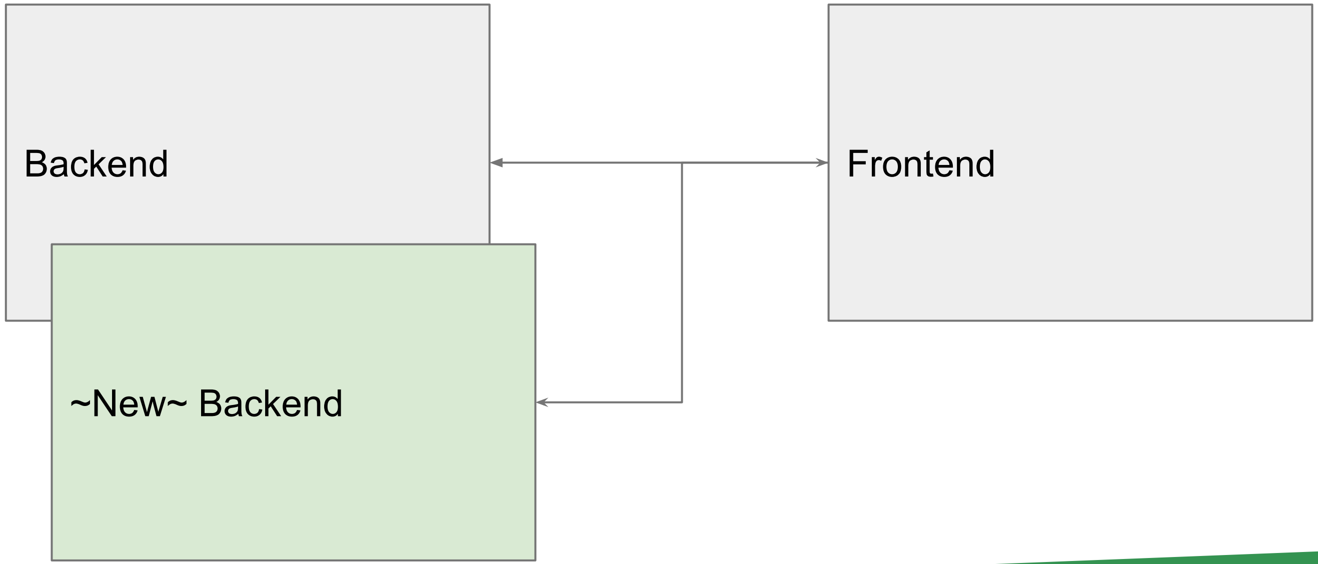
“No downtime”

- Your system will likely involve multiple different binaries



"No downtime"

- Your system will likely involve multiple different binaries



“No downtime”

Things to consider when deploying new server version

- What code is running that calls me?
- What code is running that I am calling?
- Will everything still work if I have to roll back to an old version in a few days?

“No downtime”

Things to consider when deploying new server version

- What code is running that calls me?
- What code is running that I am calling?
- Will everything still work if I have to roll back to an old version in a few days?
    - Will everything still work if traffic is split between the old and new versions for a while?

One solution: Protocol Buffer

- Key point: Define your cross-binary communication objects outside of any one binary

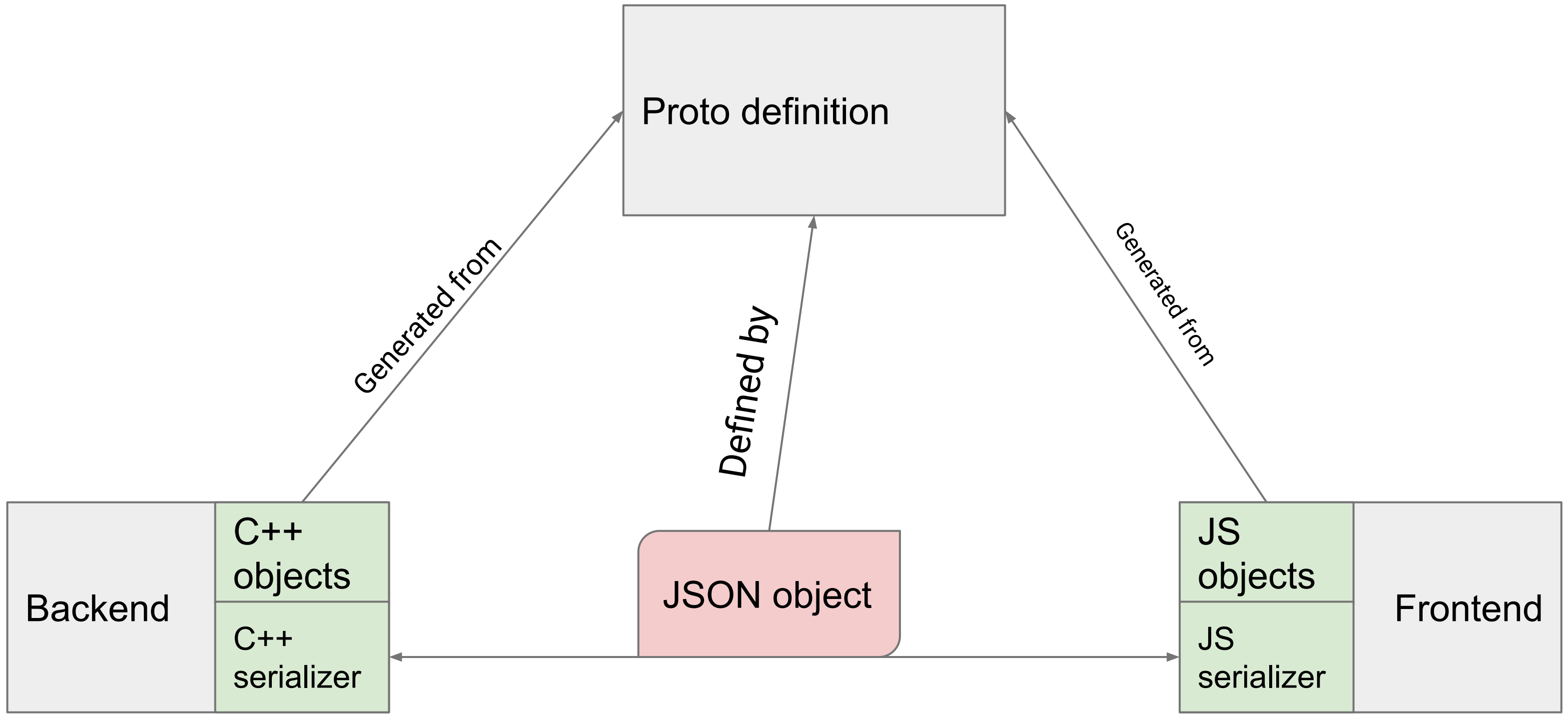
One solution: Protocol Buffer

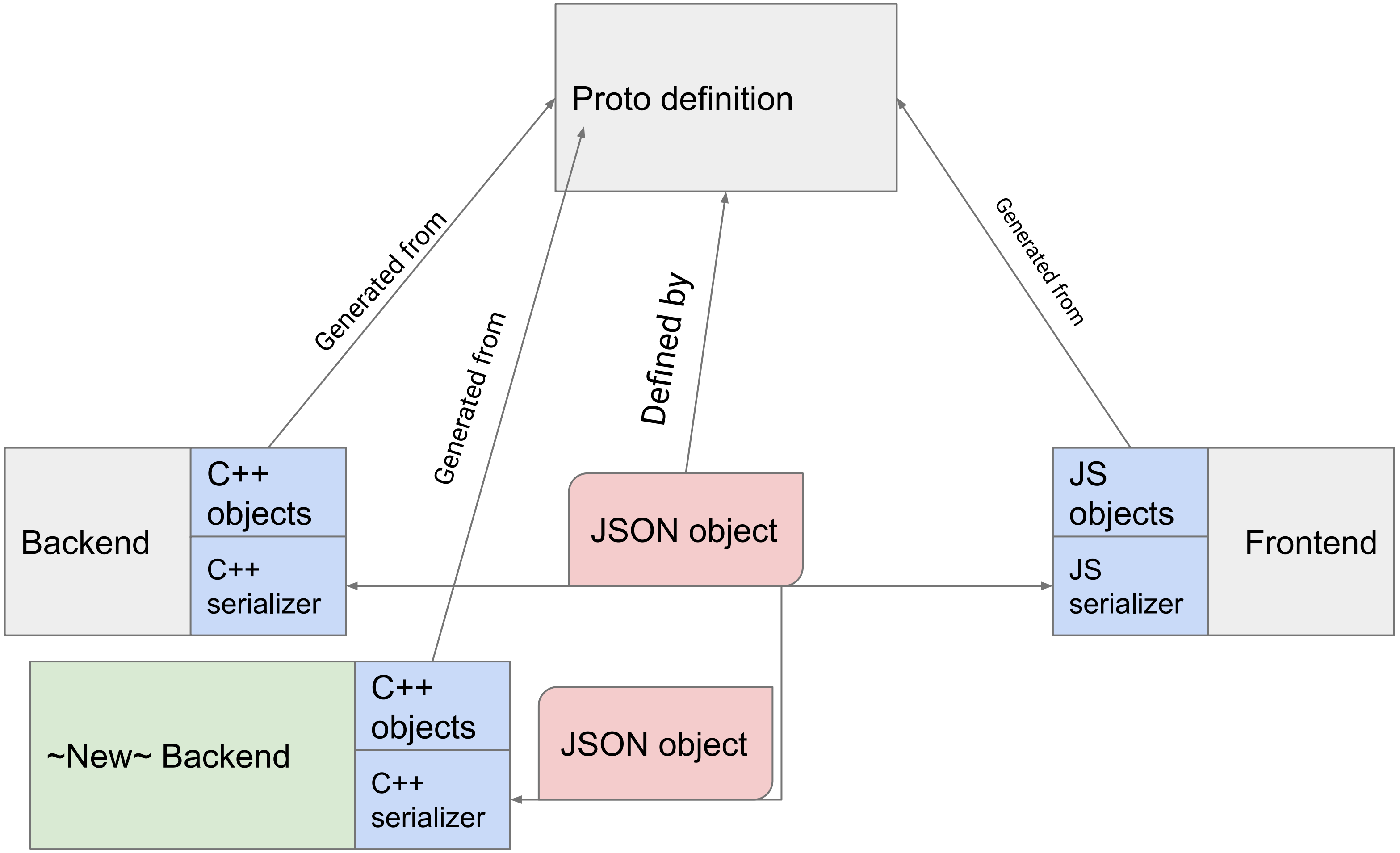
```
message Course {
  String course_id = 1;
  String department = 2;
  int credits = 3;
}
```

```
message Student {
  String name = 1;
  Date birthday = 2;
  int credits_completed = 3;
  Course favorite_course = 4;
}
```

One solution: Protocol Buffer

- Once you define the object in an implementation agnostic way, you can generate different representations of that data on the fly
    - C++ object
    - C struct
    - Haskell typeclass
    - JSON
    - Database representation
    - Etc.


Proto definition
Generated from
Defined by
Generated from
Backend
C++ objects
C++ serializer
JSON object
JS objects
JS serializer
Frontend


Proto definition
Generated from
Generated from
Defined by
Generated from
Backend
C++ objects
C++ serializer
JSON object
JS objects
JS serializer
Frontend
~New~ Backend
C++ objects
C++ serializer
JSON object

Ok, well how do I update the Proto then?

- Very carefully!
- Backwards compatibility
  - Make sure that older versions of the proto still work with the new definition

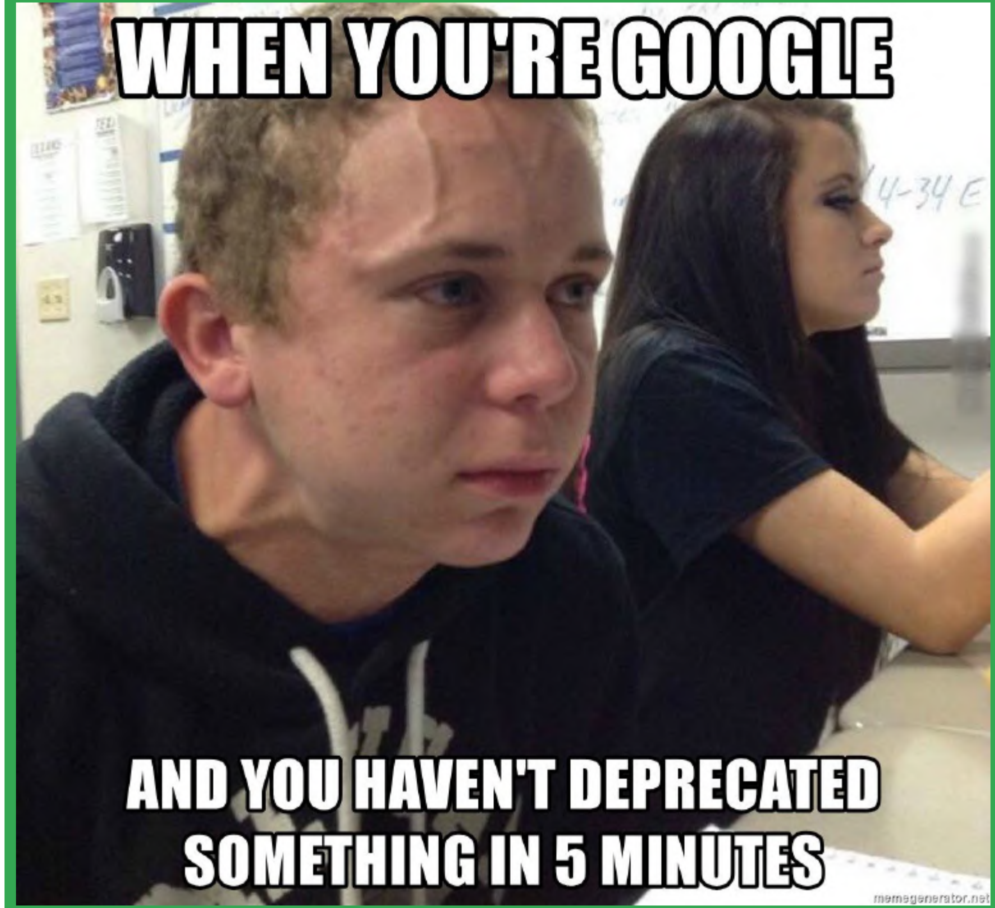
Backwards compatibility

- Sounds complicated!
- I promise it's not that bad
- There are a few operations you should avoid on a Protocol Buffer
  - Do not delete a field
  - Do not change the type of a field
  - Do not re-use IDs

Backwards compatibility

What can I do then?

- Add fields
- Mark fields as “deprecated”



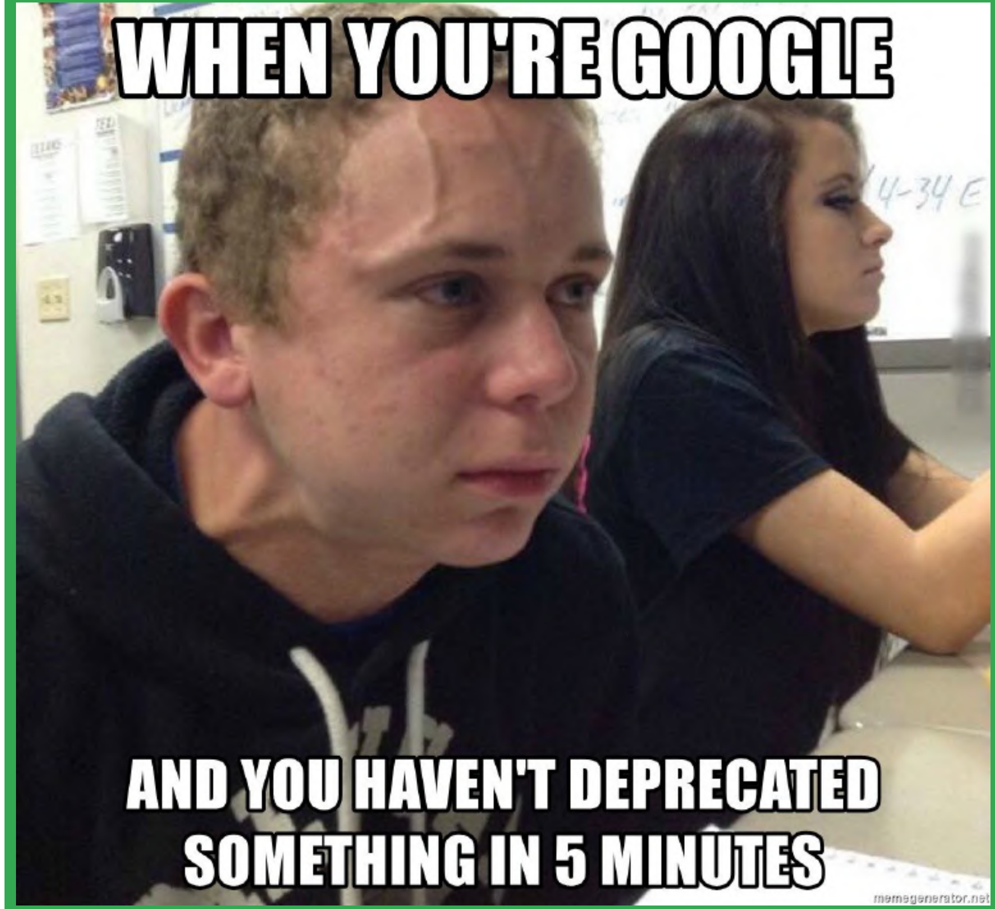
Backwards compatibility

What can I do then?

- Add fields
- Mark fields as “deprecated”

How do I modify a field?

- Add new field you want to use
- Start populating both fields (new and old)
- Remove all usages of old field
- Stop populating old field
- (Optional) Remove old field



Forwards compatibility

- Wait, but how do I make sure newer versions of the proto work with older binaries?
  1. Ignore unknown fields at deserialization time
  2. Make sure your code doesn't send new fields until all of your dependencies are updated

Configurations

Feature flags

Key problems to solve

When you're running a live service

- Don't want to wait until full features are complete to deploy to production
- Want to be able to deploy new functionality but wait to enable it until dependencies are ready
- Want to be able to turn off functionality quickly if it is breaking things

Answer: Feature Flags

- A configuration that defines when specific functionality should be enabled in your server
- Configuration should be deployment-specific
    - i.e. the configuration should be able to take on different values in devel vs staging vs production

Example:

```
if (features.enableSaving) {
  showSaveButton();
}
```

Feature Flags

- Don’t want to wait until full features are complete to deploy to production
    - Great! Set enableSaving to false in production while you build
- Want to be able to deploy new functionality but wait to enable it until dependencies are ready
    - Great! Whenever the backend implements saving, just set enableSaving to true in staging to test it

Example:

```
if (features.enableSaving) {
  showSaveButton();
}
```

Feature Flags

- Want to be able to turn off functionality quickly if it is breaking things
  - Ok so maybe one more requirement needed…

Configuration should be deployable independent of the running binary

Example:

```
if (features.enableSaving) {
  showSaveButton();
}
```

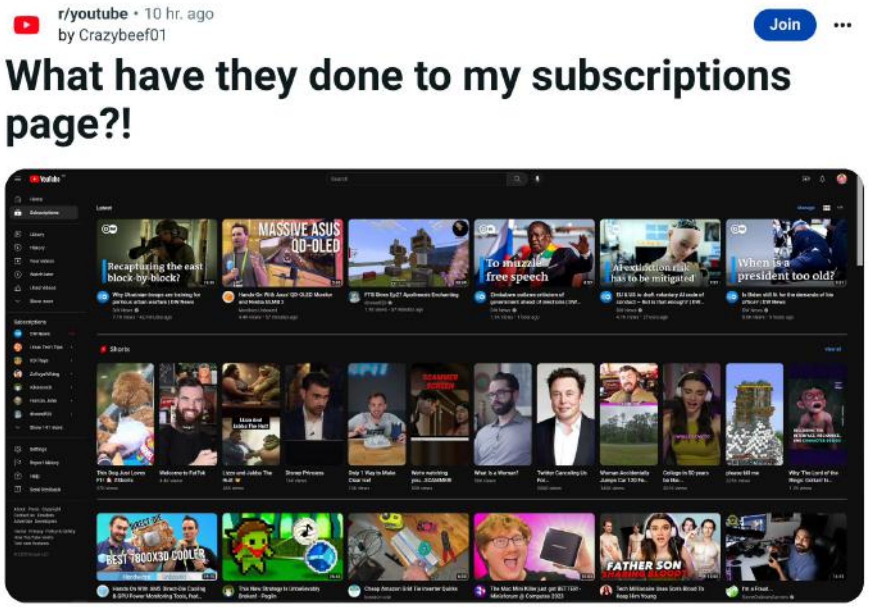
Experiments

How do we evaluate if users like our new features?

Roll them out to everyone all at once? What if there’s a bug we didn’t catch?

Roll them out to a few users at a time and see what they think : run an experiment on a percentage of your users.

Spoiler : lots of people will hate on your experiment


r/youtube • 10 hr. ago
by Crazybeef01

Join

What have they done to my subscriptions page?!

Recapturing the east block-by-block?
MASSIVE ASUS QD-OLED
To muzzle free speech
AI extinction risk has to be mitigated
When is a president too old?
Shorts
BEST 1800X3D COOLER
FATHER SON

“When I go to my subscriptions, I want to see my subscriptions, not the algorithm at the top of the page…”

“The Youtube algorithm seems pretty convinced I'm a right wing asshole interested in the opinions of other right wing assholes. No matter how many times I click not interested or skip their shorts I can't convince the algorithm im not interested.

"Stop consuming content we didn't suggest. Here, we made that a lot harder."

“Just when you thought the UI couldn’t get any worse, YT delivers.”

“I got this a month ago. The fact that they manage to do worse every single update truly is wonderful.”

“My sub page is not like this yet... I hope it isn't some slow rollout thing I'm gonna be angry if they finally came for my sub page”

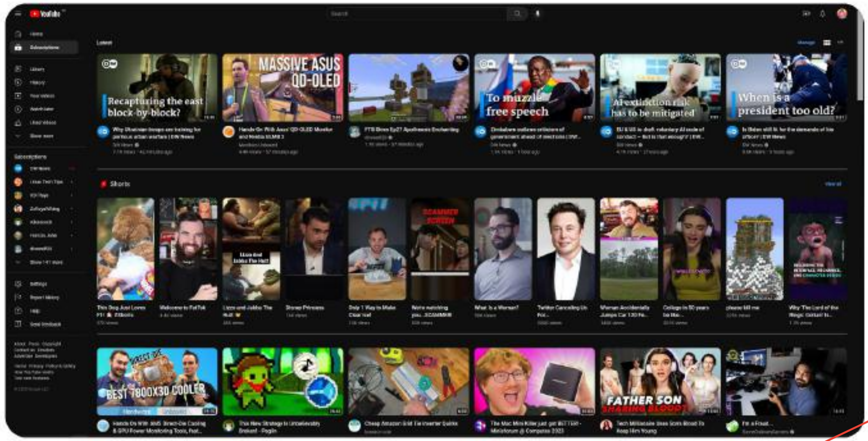
“You mean you don’t want to watch shorts instead of your subs?!?! Alphabet is lucky they don’t have any real competition or else people would have left a long time ago.”

Spoiler : lots of people will hate on your experiment

r/youtube • 10 hr. ago
by Crazybeef01

Join •••

What have they done to my subscriptions page?!



Latest
MASSIVE ASUS QD-OLED
Recapturing the east block-by-block?
To muzzle free speech
AI extinction risk has to be mitigated
When is a president too old?
Shorts
BEST 1800X3D COOLER
FATHER SON

"When I go to my subscriptions, I want to see my subscriptions, not the algorithm at the top of the page…"

"The Youtube algorithm seems pretty convinced I'm a right wing asshole interested in the opinions of other right wing assholes. No matter how many times I click not interested or skip their shorts I can't convince the algorithm im not interested.

"Stop consuming content we didn't suggest. Here, we made that a lot harder."

"Just when you thought the UI couldn't get any worse, YT delivers."

"I got this a month ago. The fact that they manage to do worse every single update truly is wonderful."

"My sub page is not like this yet... I hope it isn't some slow rollout thing I'm gonna be angry if they finally came for my sub page"

"You mean you don't want to watch shorts instead of your subs?!?! Alphabet is lucky they don't have any real competition or else people would have left a long time ago."

This person is in our experiment

Experiment Setup

Control (a.k.a. Holdback)

No change. Used to establish a baseline.

Treatment (a.k.a. Experiment)

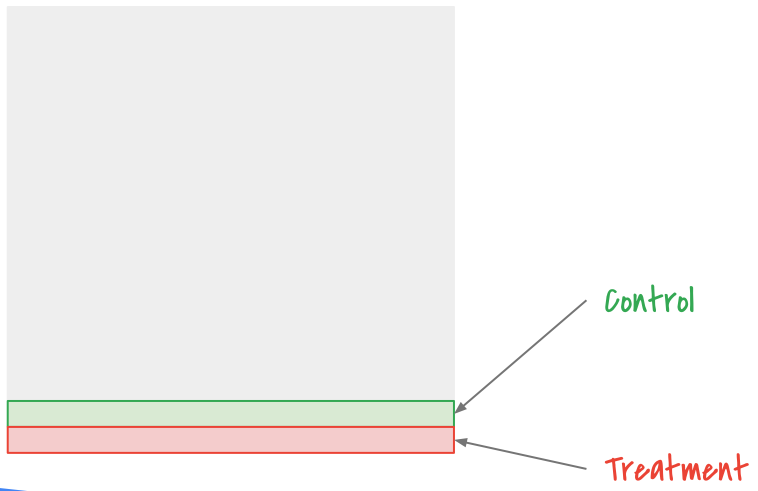
The change/feature we are interested in.

Compared to the control, what metrics have changed?

Experiment diversion

How do you select what goes into a control versus the treatment?

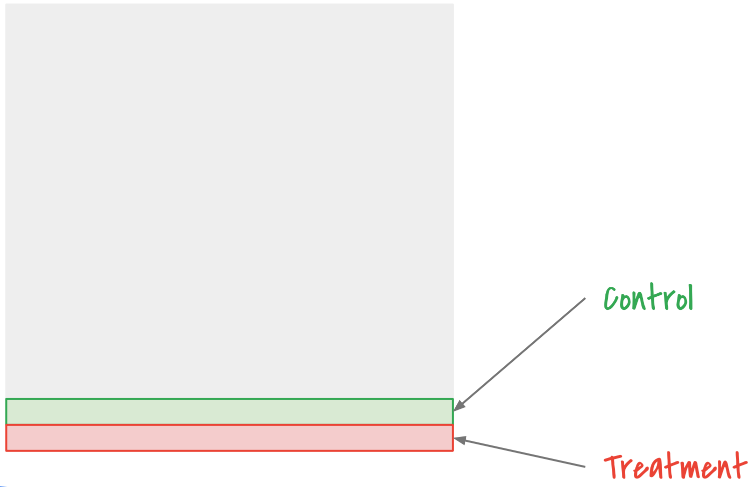
- Set aside one part of your traffic for the control (for example, 1%). Set aside a second, identically sized part of your traffic for the experiment.

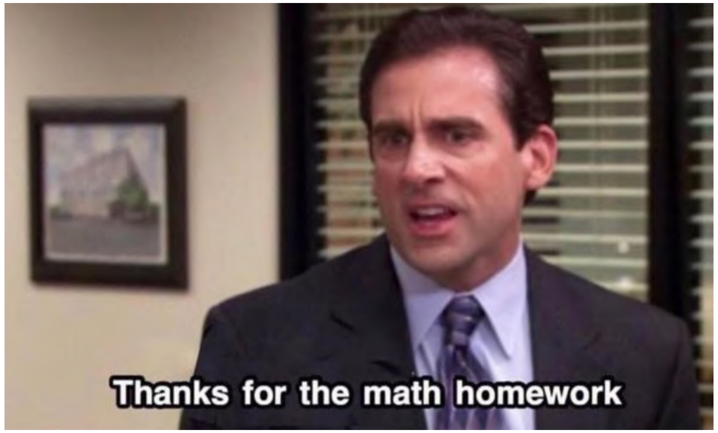


Experiment diversion

How do you select what goes into a control versus the treatment?

- Set aside one part of your traffic for the control (for example, 1%). Set aside a second, identically sized part of your traffic for the experiment.
- **Don't** : set up a 1% experiment and compare against 1/99 of your regular traffic.
    - Makes the math too hard. Also makes it impossible to calculate confidence intervals.





Experiment diversion

Example diversion criteria :

- User-level diversion.
- Video-level diversion (rare).
  - Ex : We want to experiment with changing the thumbnail of a video. Problem : we don't want to have one video have one thumbnail for 1% of the population and a different one for the other 99%.
- Request-level diversion (rarer)

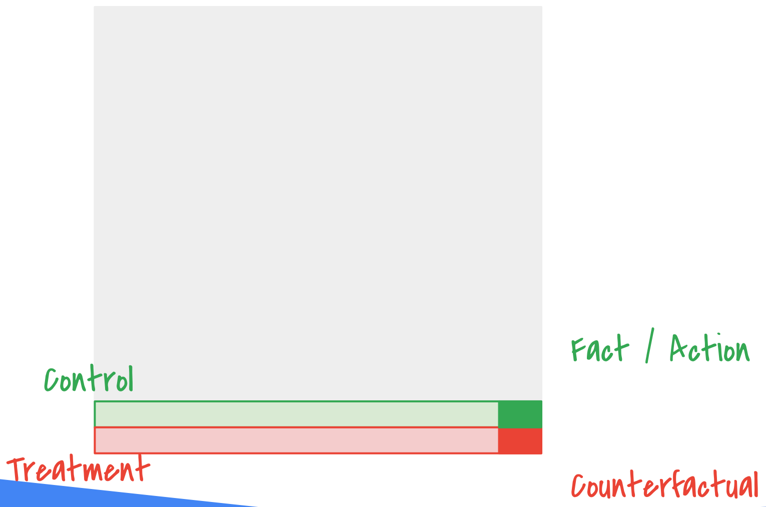
How to implement diversion?

- Cookies / sessions
- User ID
- Video ID

Counterfactuals

Really fancy way to sound smart.

- What if you want to experiment on something that happens to a particular subset of users?
- Ex : notifications. Not everyone in your 1% experiment will get the notification you're experimenting on.
- A counterfactual is a direct measurement of what **would have happened** if this action did or did not exist (i.e. a notification).
  - Slice : Only look at users who took a specific action (clicked on the subscriptions tab)
- Useful for measuring impact on small features.


Control
Fact / Action
Treatment
Counterfactual