

Self-Destructing Feature Flags

Jamie Gaskins (he/him)

[@jamie_gaskins](https://twitter.com/jamie_gaskins)



@jamie_gaskins

Agenda

- Feature Flags?
- Self-Destruct Sequence
- Tradeoffs

@jamie_gaskins

Feature Flags

@jamie_gaskins

**Run old and new code
simultaneously**

@jamie_gaskins

Deployment != Release

@jamie_gaskins

Old and Busted

New Hotness

Key/Value Store

@jamie_gaskins


```
foo:
  global: false
bar:
  ratio: 0.2
baz:
  users:
    - me
    - myself
    - i
```

@jamie_gaskins

Enable Globally

@jamie_gaskins

**Enable a percentage
of time/traffic**

@jamie_gaskins

**Enable for a
subset of users**

@jamie_gaskins

Escape Hatch

@jamie_gaskins

Bugfix Deploy

@jamie_gaskins



I have part... of a plan!



What percentage of a plan do you have?



I don't know... 12 percent.



12 percent?!! ha-ha-ha-ha.

@jamie_gaskins

Rollbacks

@jamie_gaskins

Disable Feature Flag

@jamie_gaskins

Agenda

- Feature Flags?
- Self-Destruct Sequence
- Tradeoffs

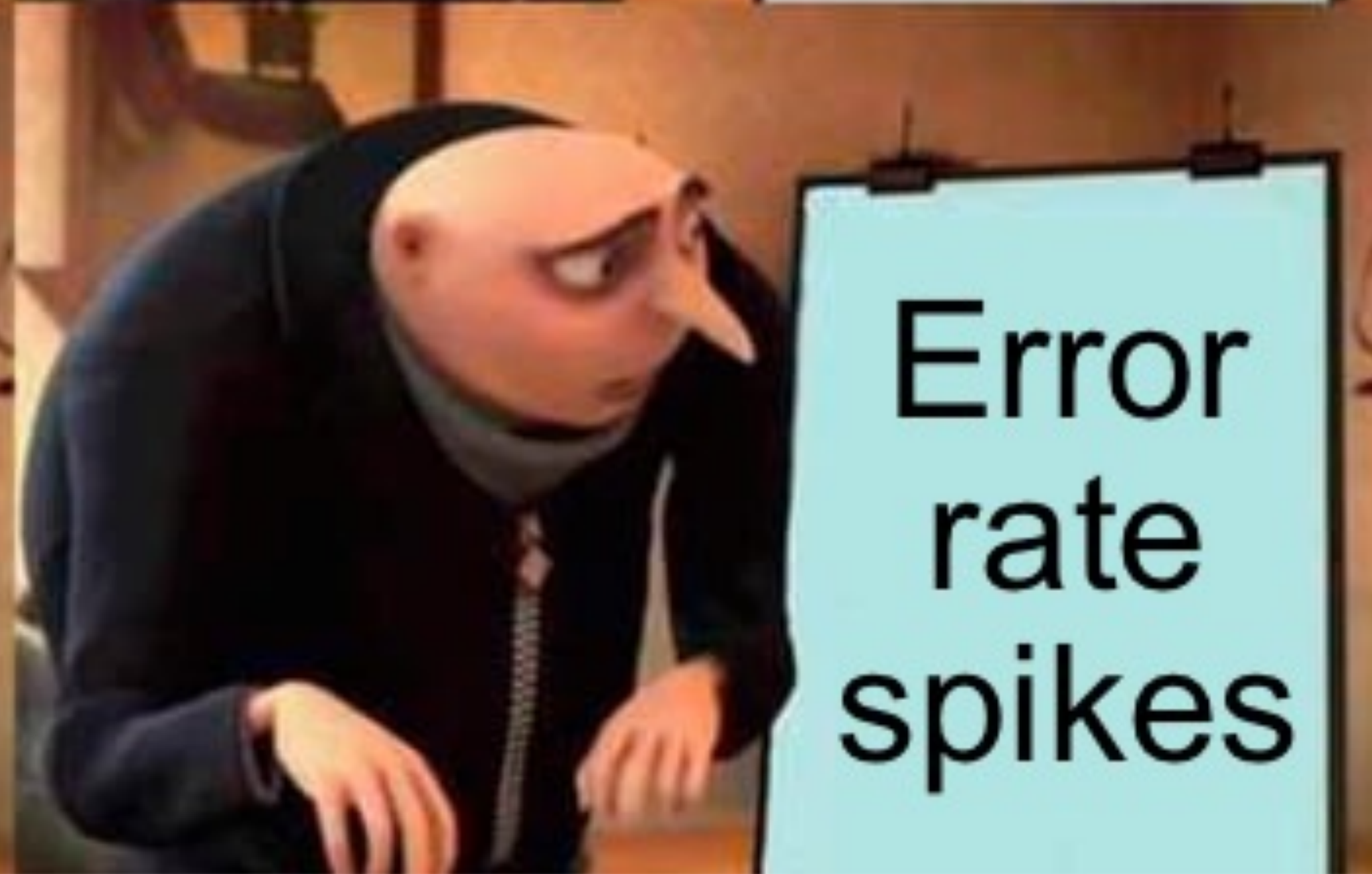
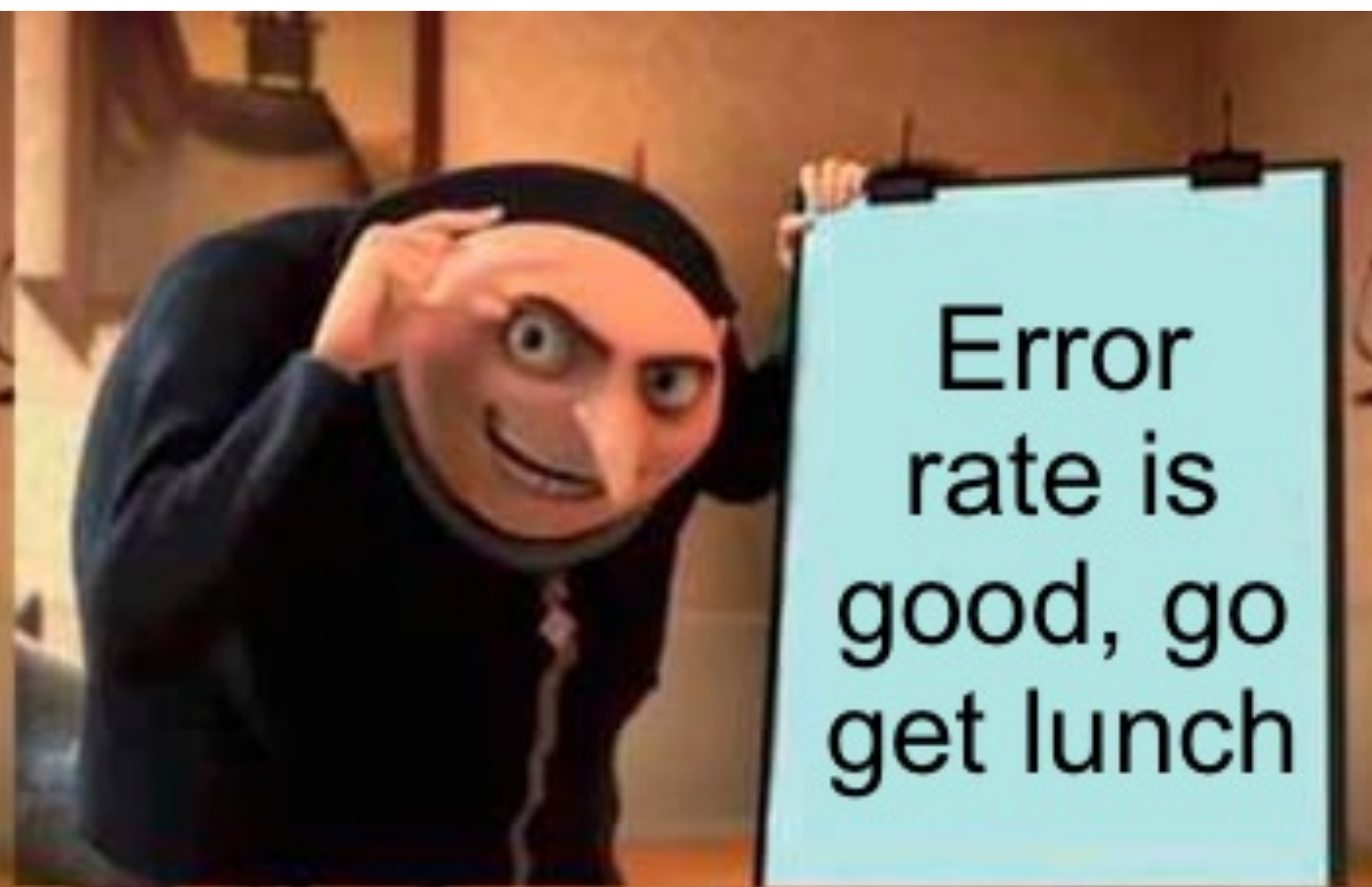
@jamie_gaskins

Self-Destructing Feature Flags

@jamie_gaskins

Scenario

@jamie_gaskins



imgflip.com

@jamie_gaskins

Remediation

@jamie_gaskins

Toil

@jamie_gaskins

TYPICAL PROCESS LATENCY:

AUTOMATED
STEPS:
800 MS

AUTOMATED
STEPS:
200 MS



SOMEONE COPIES AND PASTES DATA
FROM A THING INTO ANOTHER THING:
2-15 MINUTES
(MORE IF THE PERSON ON CALL IS BUSY)

HOWTO: Initiate self-destruct sequence

@jamie_gaskins

```
FEATURE_FOO = Feature.find("foo")
```

@jamie_gaskins

```
if FEATURE_FOO.enabled?  
  new_hotness  
else  
  old_and_busted  
end
```

@jamie_gaskins

```
if FEATURE_FOO.enabled?  
  new_hotness  
else  
  old_and_busted  
end
```

@jamie_gaskins

```
if FEATURE_FOO.enabled?  
  new_hotness🔥🔥🔥  
else  
  old_and_busted  
end
```

@jamie_gaskins

```
if FEATURE_FOO.enabled?  
  ✨new_hotness🔥🔥🔥  
else  
  old_and_busted  
end
```

@jamie_gaskins

```
CHECK_FOO = FeatureCheck.new("foo") do
  FEATURE_FOO.disable
end
```

@jamie_gaskins

```
CHECK_FOO = FeatureCheck.new(  
  "foo",  
  failure_threshold: 0.05,  
  minimum: 1_000,  
) do  
  FEATURE_FOO.disable  
end
```

@jamie_gaskins


```
if FEATURE_FOO.enabled?  
  new_hotness  
else  
  old_and_busted  
end
```

@jamie_gaskins

```
if FEATURE_FOO.enabled?  
    CHECK_FOO.check { new_hotness }  
else  
    old_and_busted  
end
```

@jamie_gaskins

```
CHECK_FOO = FeatureCheck.new(  
  "foo",  
  failure_threshold: 0.05,  
  minimum: 1_000,  
) do  
  FEATURE_FOO.disable  
end
```

@jamie_gaskins

```
foo:  
  total_checks: 12345  
  failures: 2
```

@jamie_gaskins

```
class FeatureCheck
  # ...

  def check
    checks = redis.incr("checks:#{name}")
    yield
  rescue
    fails = redis.incr("fails:#{name}")
    if (checks > minimum) && (fails.to_f / checks > failure_threshold)
      failure_handler.call
    end
    raise
  end
end
```

@jamie_gaskins

```
err := fc.Check("foo")
if err != nil {
    return nil, err
}
```

```
result, err := newHotness()
if err != nil {
    fc.Fail("foo")
    return nil, err
}
```

```
return result, nil
```

@jamie_gaskins

```
func (*fc FeatureCheck) Check(name string) error {
    _, err = fc.Redis.Incr(fmt.Sprintf("checks:%s", name))
    return err
}

func (*fc FeatureCheck) Fail(name string) error {
    fails, err = fc.Redis.Incr(fmt.Sprintf("fails:%s", name))
    if err != nil {
        return err
    }

    checks, err = fc.Redis.Get(fmt.Sprintf("checks:%s", name))
    if (checks > fc.Minimum) && (fails / checks > fc.FailureThreshold) {
        fc.FailureHandler()
    }
    return err
}
```

@jamie_gaskins

Atomic Updates

@jamie_gaskins

Account for volume

@jamie_gaskins

Distribute keys

@jamie_gaskins

Batch updates for checks and failures

@jamie_gaskins

Reset periodically

@jamie_gaskins

Agenda

- Feature Flags?
- Self-Destruct Sequence
- Tradeoffs

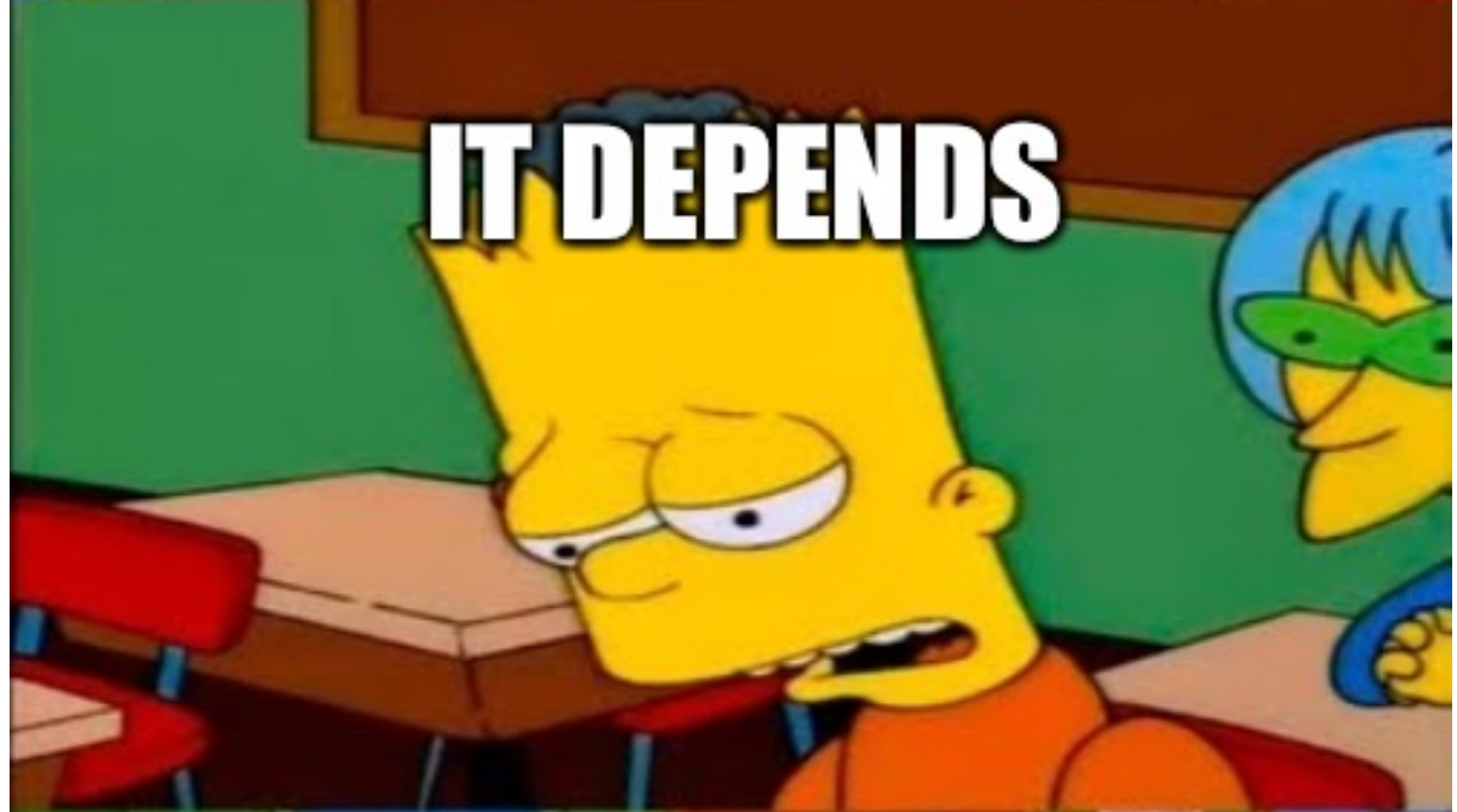
@jamie_gaskins

Tradeoffs

@jamie_gaskins



SAY THE LINE, PRINCIPAL ENGINEER



IT DEPENDS



@jamie_gaskins

Difficult to generalize

@jamie_gaskins

Long-running feature flags for beta groups

@jamie_gaskins

Requires up-front thinking

@jamie_gaskins

Traffic levels

@jamie_gaskins

Acceptable error rates

@jamie_gaskins

What are your failure modes?

@jamie_gaskins

How quickly should it react?

@jamie_gaskins

What units should be used?

@jamie_gaskins

What if you're wrong?

@jamie_gaskins

Agenda

- Feature Flags?
- Self-Destruct Sequence
- Tradeoffs

@jamie_gaskins

Recap

- Feature flags can quickly revert to a known good working version without a deploy or revert
- Disabling feature flags based on granular error rates can be automated by counting the number of checks and failures
- Think before automating it
- If you're going to automate it, use an appropriate sample size

@jamie_gaskins

Thanks, friends!

@jamie_gaskins