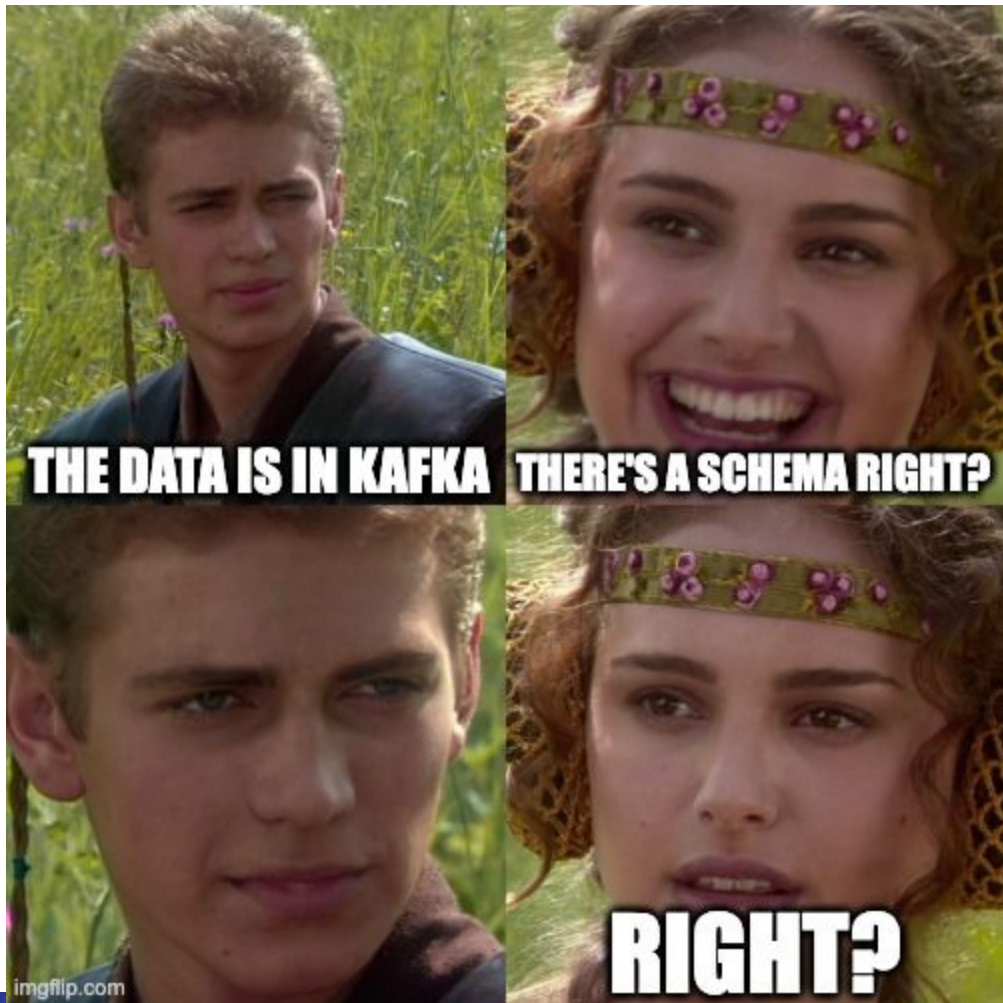
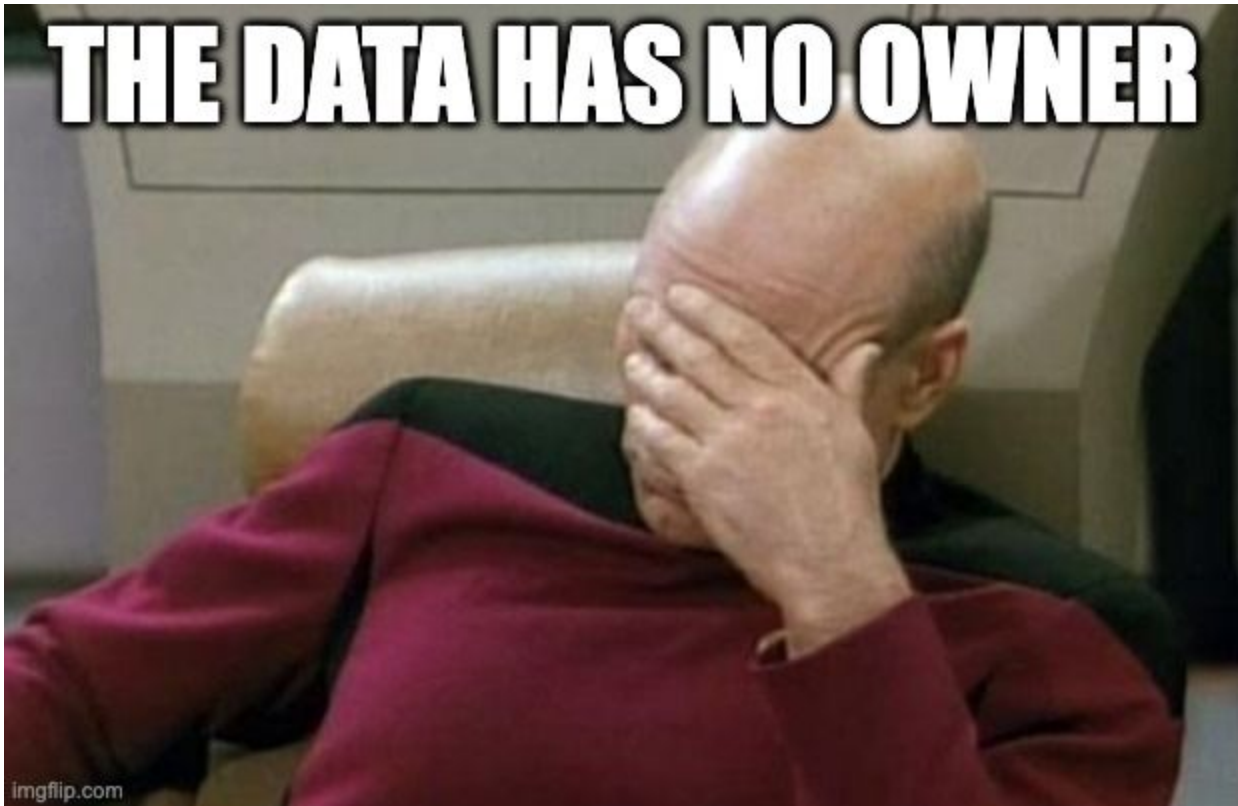


# Schema Driven Development Leveraging Open Source Tools

Brandon Brown - Senior Staff Software Engineer - SecurityScorecard



**THE DATA HAS NO OWNER**



imgflip.com

# Problems to Solve

- Share Data models between teams
- Enforce best practices for schemas
- Utilize peer review
- Support multiple languages
- Must work with or without a schema registry
- Easy to grok ownership



# Solutions Explored

- Schema formats
  - Protobuf
  - Avro
  - JSON Schema
  - No Schemas
- Registries
  - Confluent Schema Registry
  - AWS Glue Schema Registry
  - Buf Schema Registry
  - Apicurio Schema Registry



# Exploration Results - Schema Formats

Format	Multi Language	Non Developer Friendliness	Validation Tooling
Protobuf			
Avro			
JSON Schema			
No Schemas			

# Exploration Results - Registries

Registry	Formats
Confluent Schema Registry	Protobuf, Avro, JSON Schema
AWS Glue Schema Registry	Protobuf (wasn't supported at evaluation time), Avro, JSON Schema
Buf Schema Registry	Protobuf
Apicurio Schema Registry	Protobuf, Avro, JSON Schema

# The Winner - Schemas

- Protobuf
  - IDL based
  - Compile to different languages
  - Robust Tooling
  - Compatibility check at the schema level





# The Winner - Registry

- Confluent
  - Supported all the formats
  - Seamless Integration with Kafka
  - Library support out of the box
  - Easy work around for handling messages in the case of no registry



# The Solution

- Git Repository with 2 branches
- OmniProto for Proto compilation
- Buf CLI for linting and compatibility check
- Artifactory for library hosting
- Jenkins for ci/cd
- Bash to tie it all together



# Omniproto

- <https://github.com/grpckit/omniproto>
- Toolkit for targeting multiple language targets
- Yaml configuration
- Solves the problem of manually scripting for each language



# Omniprototo Example

```
rootdir: protobuf
sources: # all sources will be passed together
  - com/bbrownsound/proto
output: gen # gen is the default output directory (should be gitignored)
plugins:
  - name: python
    output: python-protobuf-models/src/bbrownsound
  - name: cpp
    output: cpp-protobuf-models/src
  - name: go
    args: paths=source_relative
    output: go-lang-protobuf-models
  - name: js
    output: gen/js
  - name: java
    output: scala-protobuf-models/src/main/java
  - name: scala
    args: flat_package,java_conversions
    output: scala-protobuf-models/src/main/scala
descriptors:
  output: gen/descriptors.pb # defaults to descriptors.pb in root output directory
  enabled: true # defaults to true, descriptors are useful
  include_imports: true
  include_source_info: true
debug: true # debug will output command line information
dry_run: false # if true, don't execute the protoc command. useful with debug.
```

# Buf

- Linting
- Compatibility Checking
- Why use Buf for checking?
  - Doesn't require registry to do the check (pro and a con)
  - Easy to use
  - Confluent tooling would require setting up a java project



# Buf Example

```
version: v1
lint:
  use:
    - MINIMAL
    - FILE_LOWER_SNAKE_CASE
    - ENUM_VALUE_UPPER_SNAKE_CASE
    - ENUM_FIRST_VALUE_ZERO
breaking:
  use:
    - WIRE
```

# How does it work?

- Set up stubbed directories for each language
- gitignore the output directories
- Run omniproto
- Publish to your repository of choice
- Codeowners file + Package Structure



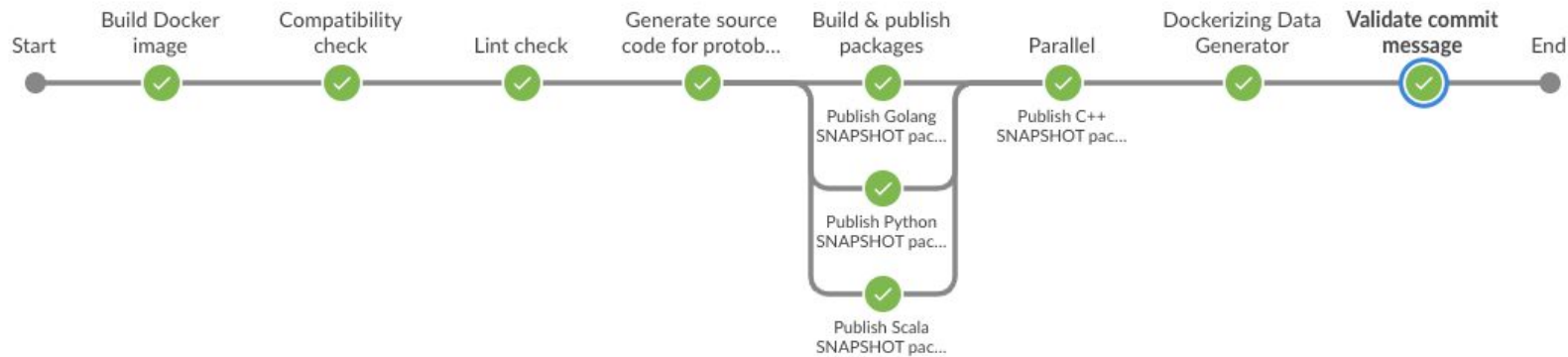
# Example Build Failure

No changes  
Push event to branch DP-383





# Example Build Success



# Lessons Learned

- Lack of direct integration with the schema registry can lead to unknowns
- Versioning with golang is hard
- Providing example projects is key to onboarding
- Documentation is king
- Enforce standards and they will come
- Determining when to make releases is tricky
- Data as a first class citizen helps design decisions



**I DON'T  
ALWAYS SCHEMAS**



**OH WAIT, I DO**

# Questions?



# Connect with Me

- LinkedIn: <https://www.linkedin.com/in/brandon-brown-8529681b/>
- Github: <https://github.com/brbrown25>
- Twitter: @brbrown
- Code Samples: <https://github.com/brbrown25/current-23-talk>

