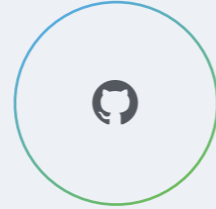




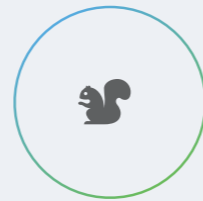
GraphQL at GitHub



👋 **Hi, I'm @rmosolgo**

- graphql-ruby: Feb 2015
- GitHub: May 2017
- Netherlands: Monday

Why GraphQL?



USABILITY



ISOLATION



CONSISTENCY

Usability: enable clients to build things that *just work*, quickly! Self-documenting, discoverable API

Isolation: abstraction between data and view logic so we can change the DB (e.g., migrate between databases)

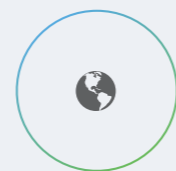
Consistency: one codepath for API and views

How GraphQL?



RUBY GEMS

- graphql
- graphql-client
- graphql-batch

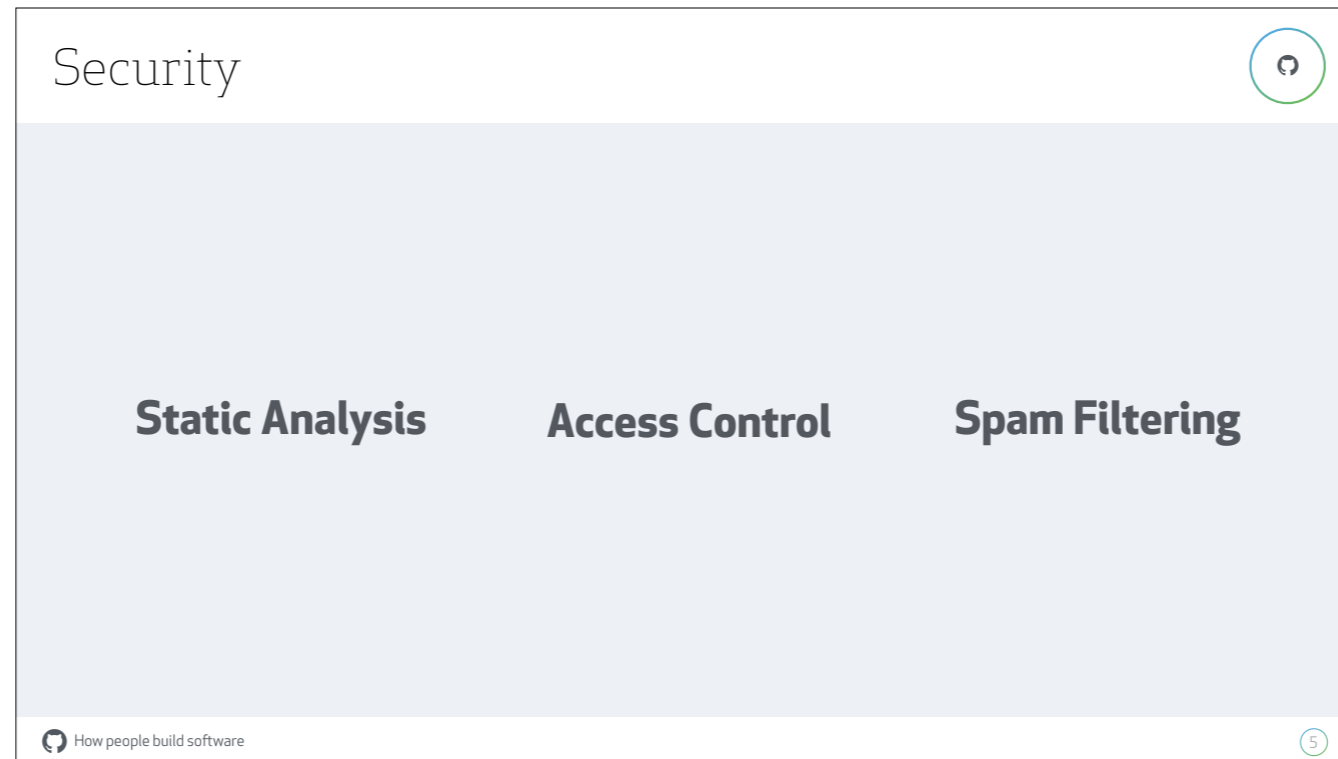


CLIENTS

- Views
- GraphQL API
- REST API

Stack is mostly open-source

Used for [GitHub.com](https://github.com) views, Rest & GraphQL APIs, and Atom integration uses Relay



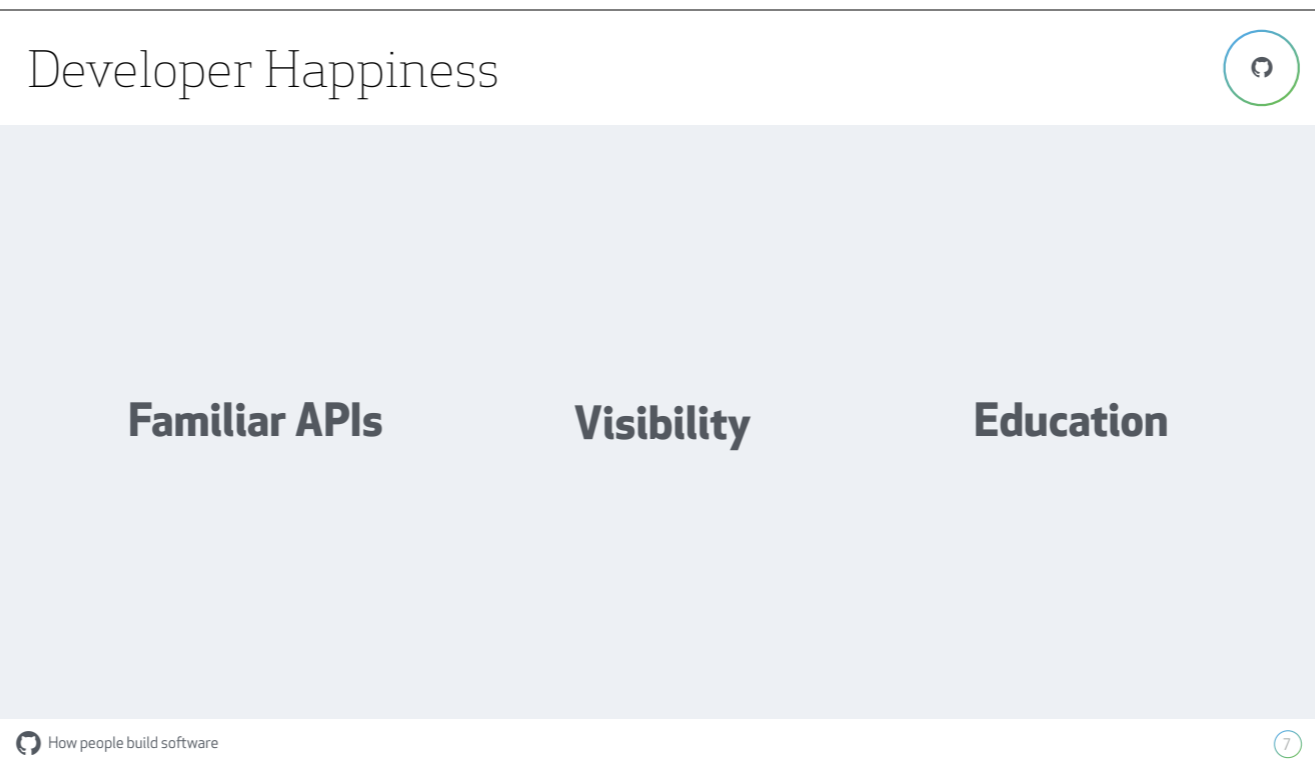
GraphQL's complexity and flexibility is scary! We manage it with:

- Analyze incoming queries for authorization and server load (rate limiting)
- Access control checks for every object
- Unified, consistent spam filtering



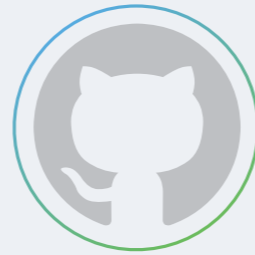
Unlike Facebook, our API is user-facing, but we're still going to change it.

- Opt-in previews for new features, use a header to reveal new data
- Publicize additions and deprecations, publish a changelog
- Help people's code still work: observe usage, contact clients, re-write queries



Help developers be productive

- Familiar APIs: something like Shopify's class-based wrapper
- Visibility: help developers understand and debug unfamiliar processes (e.g. Tracing)
- Education: Help onboard people, introduce new concepts, best practices, "pit of success"



Questions?

Robert Mosolgo
@rmosolgo
Amsterdam.rb, Aug 25 2017