AMAZEENARS

WEBINARS

PRESENTED BY

Amazee Labs

GRAPHQL & TWIG

SOFT-DECOUPLED DRUPAL

PHILIPP MELAB

- Senior Developer @ Amazee Labs Zurich
- Co-Maintainer of the GraphQL module(s)
- Located in Austria
- `pmelab` everywhere

INTRO TO GRAPHQL

- Initiated by Facebook
- Language for operations in data graphs
- Implementation agnostic

EXAMPLE QUERY

```
query {
    node:nodeById(id: "1") {
        title:entityLabel
        related:relatedNodes {
            title:entityLabel
        }
    }
}
```

EXAMPLE RESPONSE

```
{
  "node": {
    "title": "Article A",
    "related": [
      { "title": "Article B" },
      { "title": "Article C" }
    ]
  }
}
```

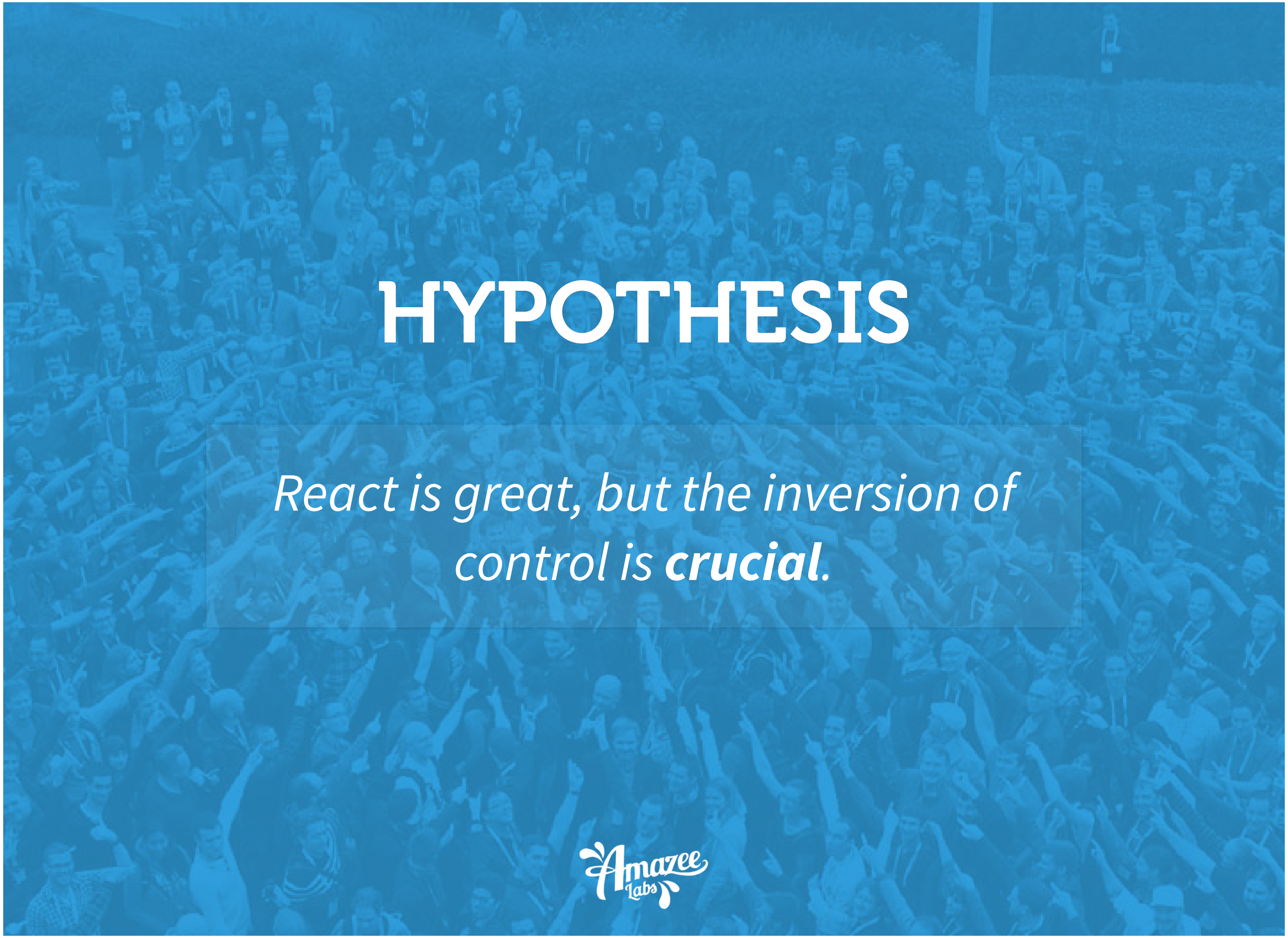
THE NEED FOR DECOUPLING

STANDARD DRUPAL

1. Accept request
2. Build response
3. Push through theme

DECOUPLED DRUPAL

1. React on user interaction
2. Request data from backend
3. *Process the request* (GraphQL)
4. Update display



HYPOTHESIS

React is great, but the inversion of control is ***crucial****.*

INCONVENIENT TRUTH

- Product development leads
- Technology follows

GOING *FULL* DECOUPLED?

- *Double* the amount of knowledge
- *Double* the amount of errors
- No Drupal forms
- No interface translation
- No other nice everyday features

THE SOLUTION

BABYSTEPS

Amazee Labs

GRAPHQL IN TWIG

```
{#graphql
query {
    users:userQuery {
        count
    }
}
#}
<p>This website is the home of {{ graphql.data.users.count }} users.</p>
```

WHAT DO I NEED?

- Composer enabled Drupal 8
  https://github.com/drupal-composer/drupal-project
- The `graphql` module
  ```
  composer require drupal/graphql
  ```
- The `graphql_twig` module
  ```
  composer require drupal/graphql_twig
  ```

LIVE DEMO!

SUMMARY

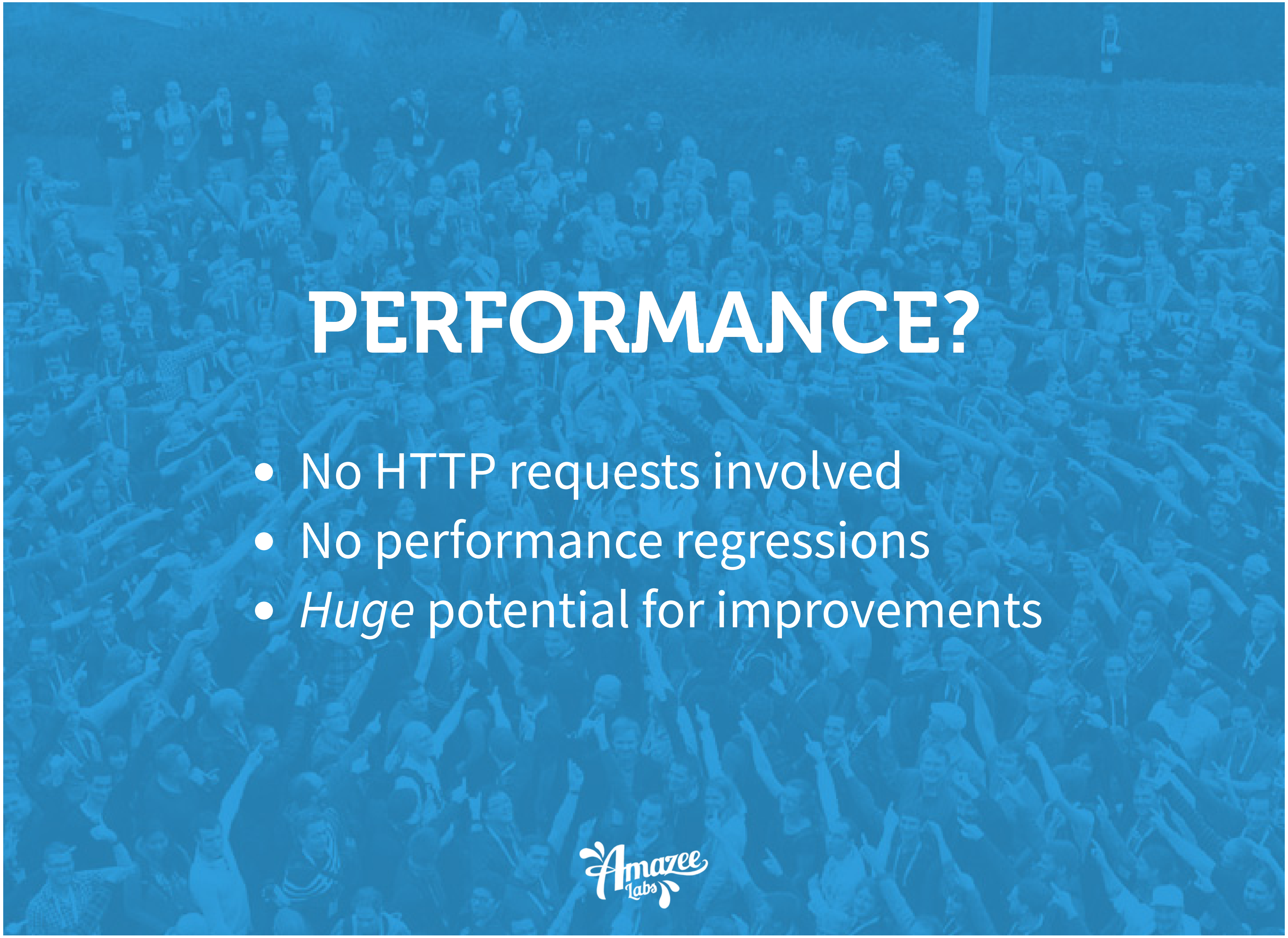
- Attach to any Twig template
- Assemble query from includes
- Match theme variables to query arguments
- Works alongside standard Drupal

BENEFITS

- Decoupled workflow
- Vertical Slicing
- Reduced risk
- Future proof

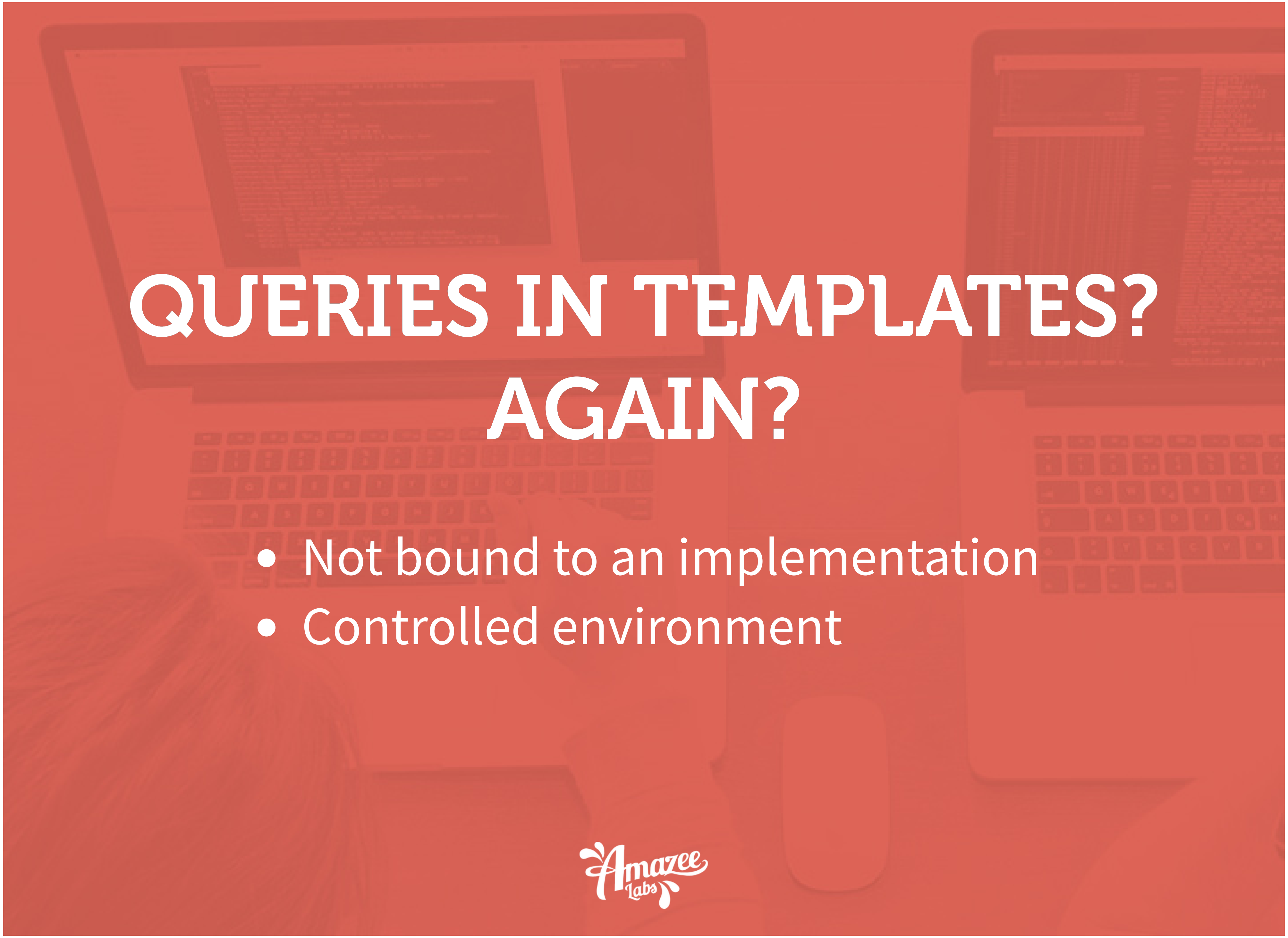
FORMS AND MUTATIONS?

- Drupal forms still intact
- Javascript widgets & GraphQL



PERFORMANCE?

- No HTTP requests involved
- No performance regressions
- *Huge* potential for improvements



QUERIES IN TEMPLATES? AGAIN?

- Not bound to an implementation
- Controlled environment

ARE WE DECOUPLED?

NOT QUITE THERE YET …

LANGUAGE BETWEEN FRONTEND AND BACKEND ...

... WITH LOTS OF DRUPALISMS.

```
query {
  nodeQuery(filters: {
    conditions: [{ "field": "field_tags", "value": ["3"]}]
  }) {
    entities {
      ... on NodeArticle {
        entityLabel
        body
      }
    }
  }
}
```

SCHEMA DEFINITION LANGUAGE

```
type Query {
  articlesByTag(id: String!): [Article]
}

type Article {
  title: String!
  body: String!
  tags: [String]
}
```

OPTIMIZED SCHEMA

```
query {
  articlesByTag(id: "3") {
    title
    body
    tags
  }
}
```

GRAPHQL SDL

- Contract between frontend and backend
- Consumed by Drupal
- Interface to mix & match existing resolvers

LONG TERM GOAL?

- **Full** decoupling within Drupal
- Dedicated theme engine
- Frontend-framework for interactive elements

QUESTIONS?

Amazee Labs