

# GraphQLConf

## 2026

hosted by  GraphQL  
Foundation

# GraphQLConf 2026

hosted by  GraphQL  
Foundation

# Inverse Conway Maneuver with GraphQL

Sam Deng, Zillow Group

#GraphQLConf

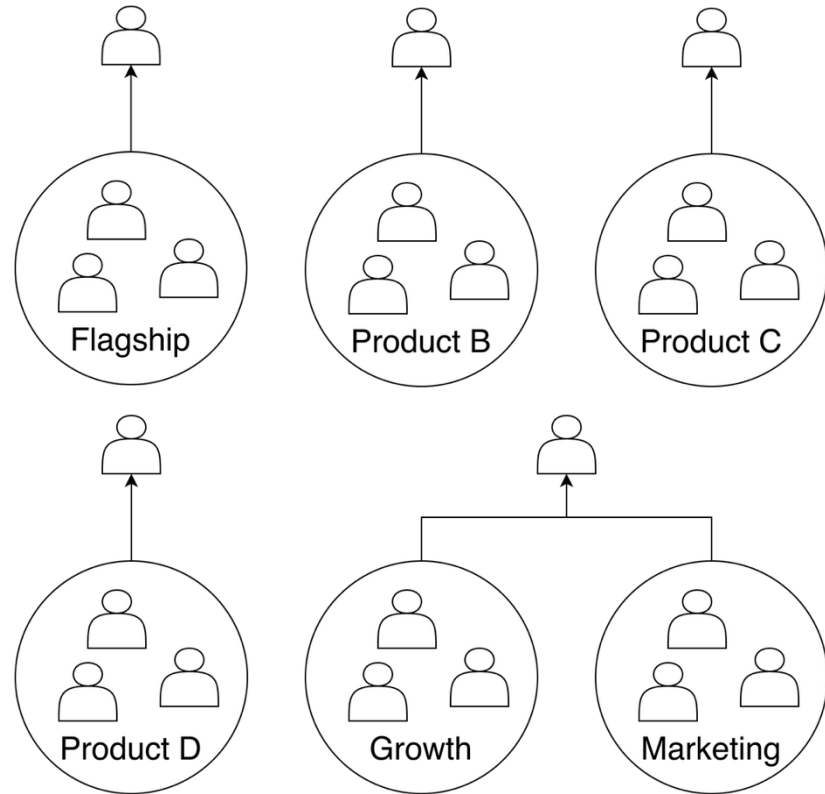
# Conway's Law

*“Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.”*

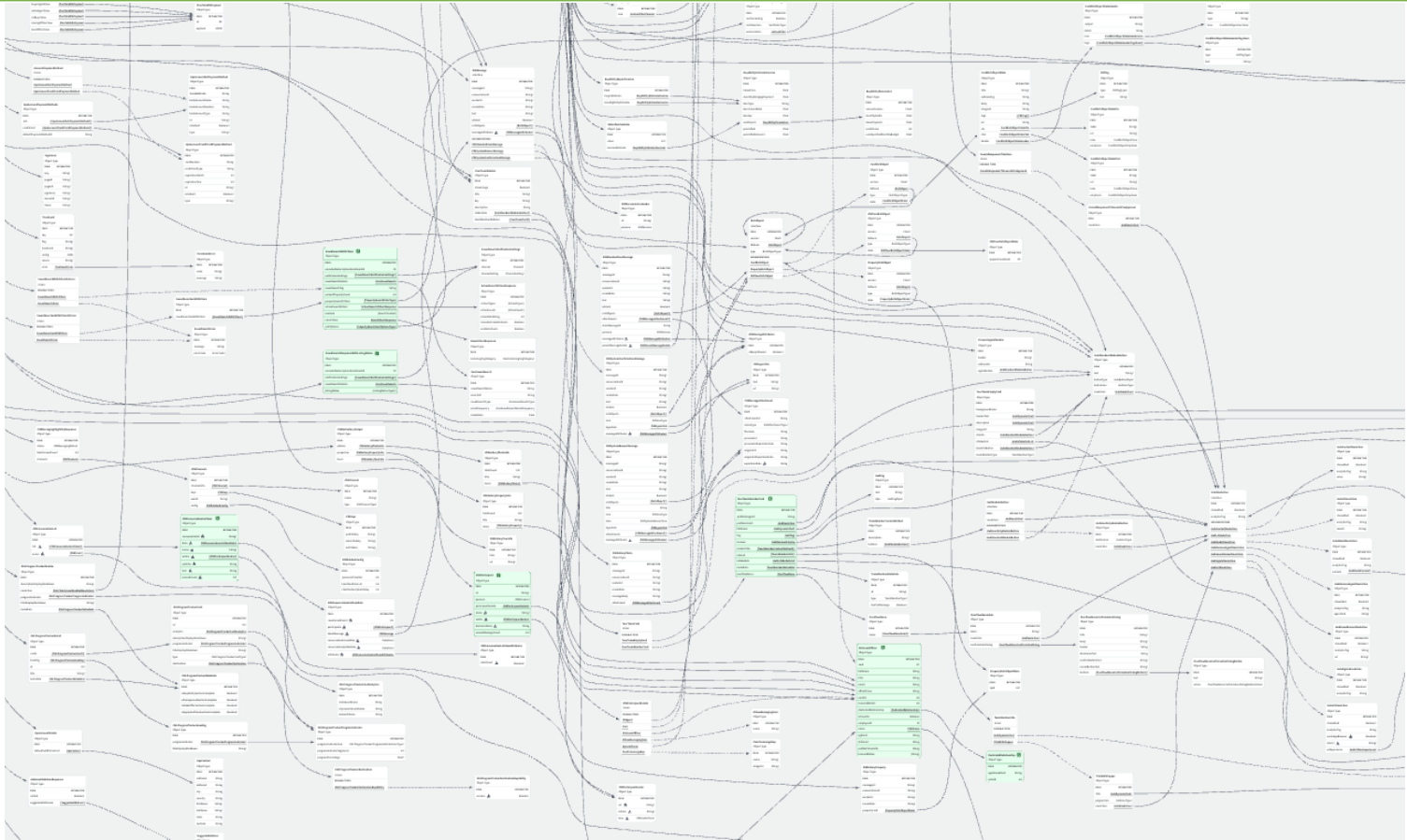
—Melvin E. Conway, How Do Committees Invent?

# Business Growth

1. Single product
  - Single team
2. More products
  - Multiple teams
3. ???
  - Disjointed architecture
  - Disjointed products
4. Profit



# Entity Treasure Hunt



# Tragedy of the Anticommons

**THE TRAGEDY OF THE  
ANTICOMMONS**  
TOO MANY OWNERS, TOO LITTLE USE.

WHEN EVERY OWNER HAS A VETO, NOTHING GETS DONE.

THE COSTS ARE REAL.

-  HIGH TRANSACTION COSTS
-  DELAY AND INACTION
-  WASTED POTENTIAL
-  MISSED OPPORTUNITIES FOR EVERYONE

**A BRIDGE THAT COULD CONNECT US ALL...  
...BUT CAN'T.**

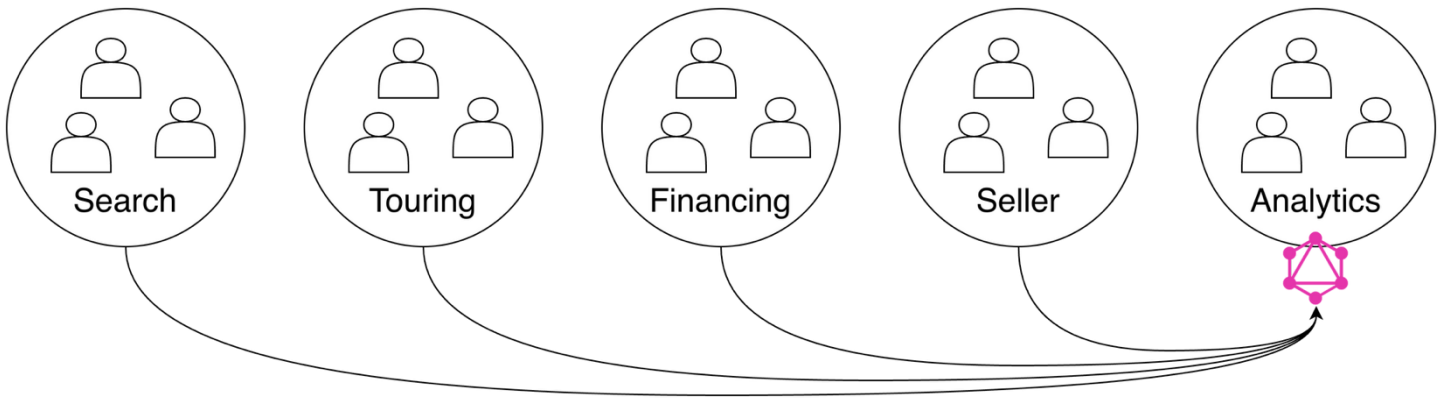
OWNER NO. 1 SAYS NO  
OWNER NO. 2 OBJECTS  
OWNER NO. 3 DEMANDS CHANGES  
OWNER NO. 4 WITHHOLDS PERMISSION  
OWNER NO. 5 WANTS COMPENSATION  
OWNER NO. 6 REFUSES APPROVAL  
OWNER NO. 7 PROHIBITS USE  
OWNER NO. 8 HAS OTHER PLANS  
OWNER NO. 9 SAYS NOT NOW

**MANY RIGHTS. NO USE. EVERYONE LOSES.**

# Data Stitching

We tried stitching together the data post-hoc...

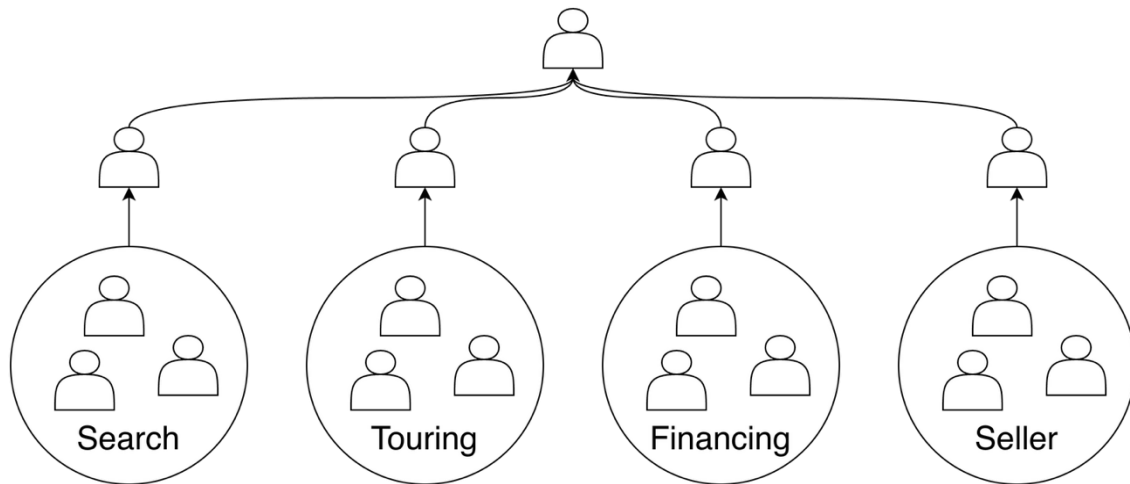
- Duplicate data creates ambiguity
- Another schema to maintain
- Sources of truth unclear



# Inverse Conway Maneuver

Why not re-org towards the desired architecture?

- Single threaded leader
- Streamlined communication & architecture



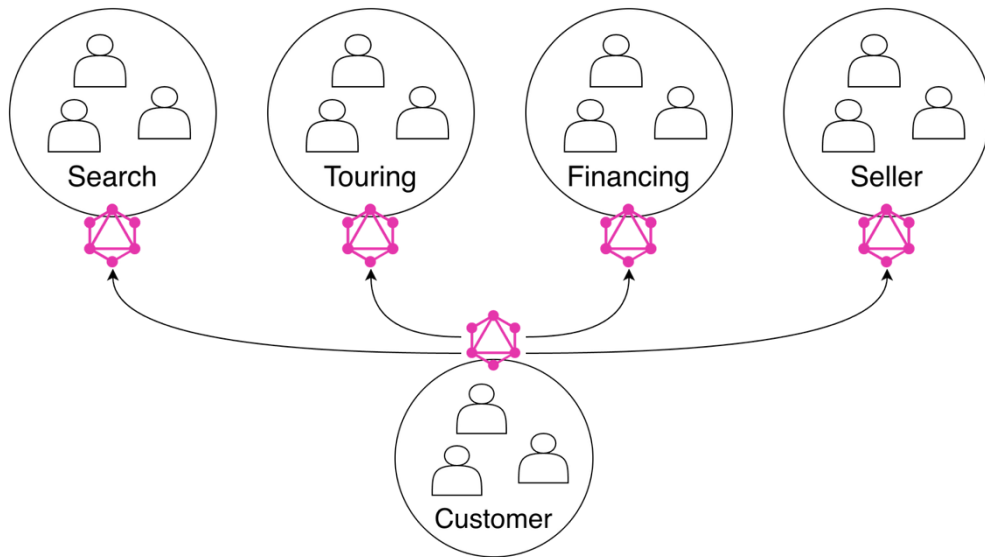
# Inverse Conway Maneuver



# Inverse Conway Maneuver

We re-organized the schema...

- Build outwards from customer identity
- Connect data producers into it



# GraphQL Schema

```
# move.graphql
type Move {
  id: ID!
  search: [Search]
  tour: [Tour]
  loan: Loan
  listing: Listing
}

extend type Loan @key(fields: "id") {
  id: ID! @external
}
```

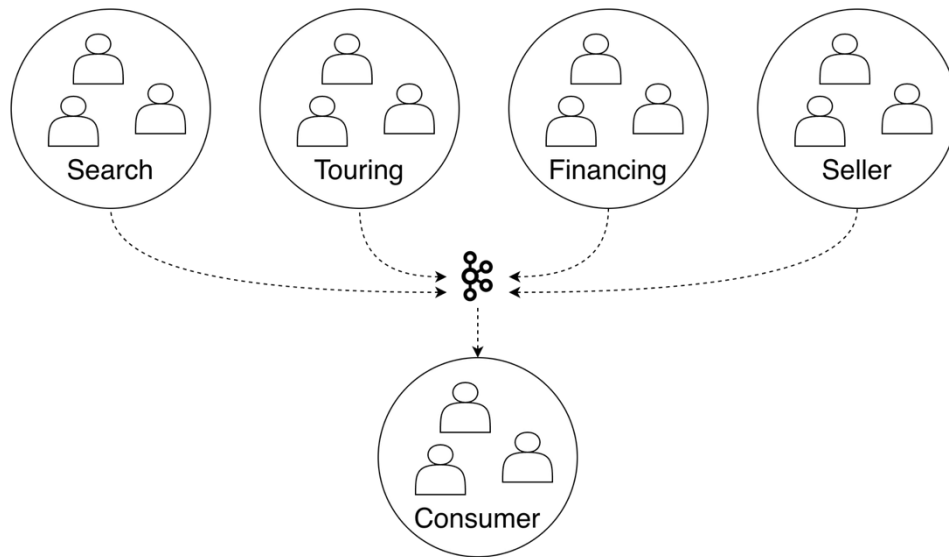
```
# financing.graphql
type Loan @key(fields: "id") {
  id: ID!
  ...
}
```

```
# Query
move(moveId: String!) {
  id
  search {...} # Owned by Search team
  tour {...} # Owned by Touring team
  loan {...} # Owned by Mortgage team
  listing {...} # Owned by Seller team
}
```

# About the Write Path

Garbage in, garbage out

- Gate which entities that link to a Move
- Standardize ingress schema



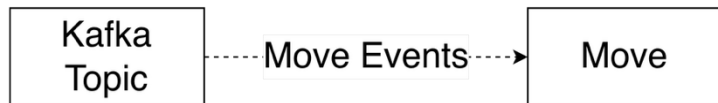
# Developer Experience

## Attaching an entity to a Move

- Managed ingress via Move SDK



- Index entity ID on write



### # Query

```
moveByCustomerId(customerId: String!)
```

```
moveByEntityId(entityId: String!, entityType: EntityType!)
```

```
moveById(moveId: String!)
```

# Thank You

What did we solve?

- Analytics & AI has better input data
- Product integrations unblocked

How did we solve it?

- Canonical map of customer journey
  - Built a single source of truth
  - Paved the write paths
- No reorgs
  - Low coupling; high alignment required

