Polymarket Historical Data Collection Strategy

Building Custom Subgraph Infrastructure for Blockchain Prediction Markets

Presenter: Zhikun Chen

November 10, 2025

Outline

Problem: Need Comprehensive Polymarket Historical Data

Research Objectives

- **Event Correlation Analysis**
  - Market price movements vs. real-world events
  - Information propagation patterns
- **User Group Identification**
  - Whale traders vs. retail participants
  - Arbitrageurs and market makers
  - Information traders vs. noise traders
- **User Action Analysis**
  - Trading patterns and strategies
  - Position management behaviors
  - Resolution participation patterns

Data Requirements

Complete historical record of **all trades**, **positions**, and **resolutions** from Polymarket launch (2020) to present

Reality: Core Data Lives On-chain (Polygon)

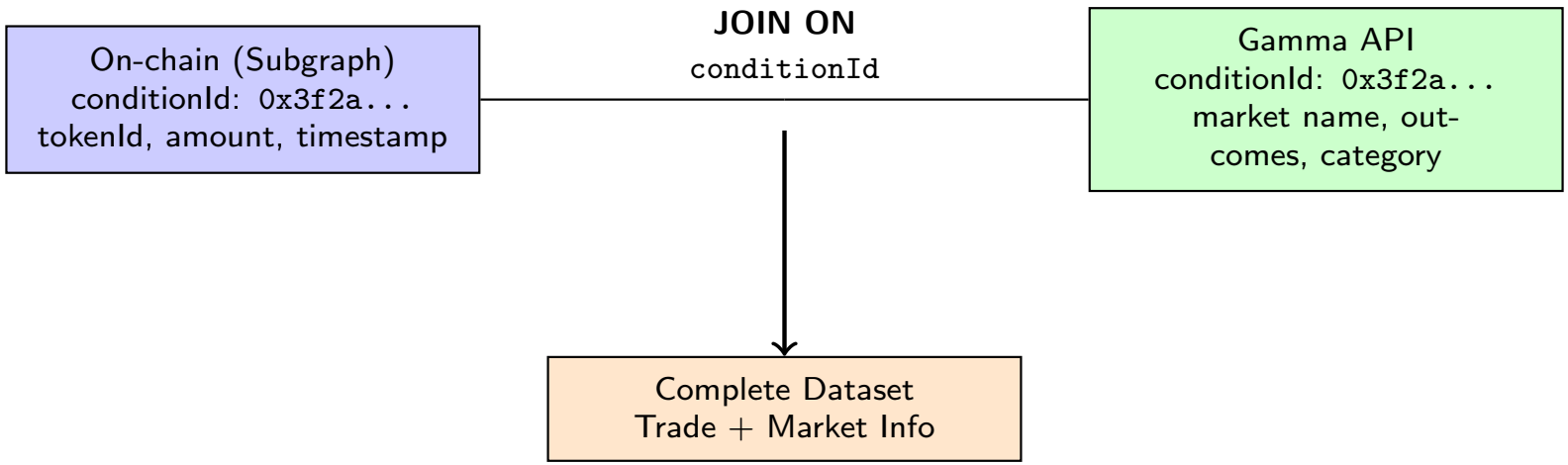
On-chain Data (Polygon PoS)

- ✓ Executed trades (fills)
- ✓ Position lifecycles
- ✓ Market resolutions
- ✓ Wallet interactions
- ✓ USDC settlements

Off-chain Data

- × Order book depth
- × Cancelled orders
- × Market metadata
- × Bid-ask spreads

Data Integration Architecture



Three Approaches to Access On-chain Data

Method	Setup Complexity	Storage Needs	Customization	Dev Time	Maintenance
Own Polygon Archive Node	High	3-5 TB	Full	2-3 weeks	Daily
Existing Subgraph	None	None	None	Immediate	None
Custom Subgraph	Medium	$<$ 100 GB	High	3-5 days	Weekly

Archive Node

+ Direct chain access
- Complex state sync
- Handle reorgs manually
- 24/7 monitoring

Existing Subgraph

+ Zero engineering
- Black box logic
- Cannot modify schema
- Missing custom metrics

Custom Subgraph

+ Tailored analytics
+ Version control
+ Automated indexing
+ Built-in resilience

Custom Subgraph (Optimal Engineering Solution)

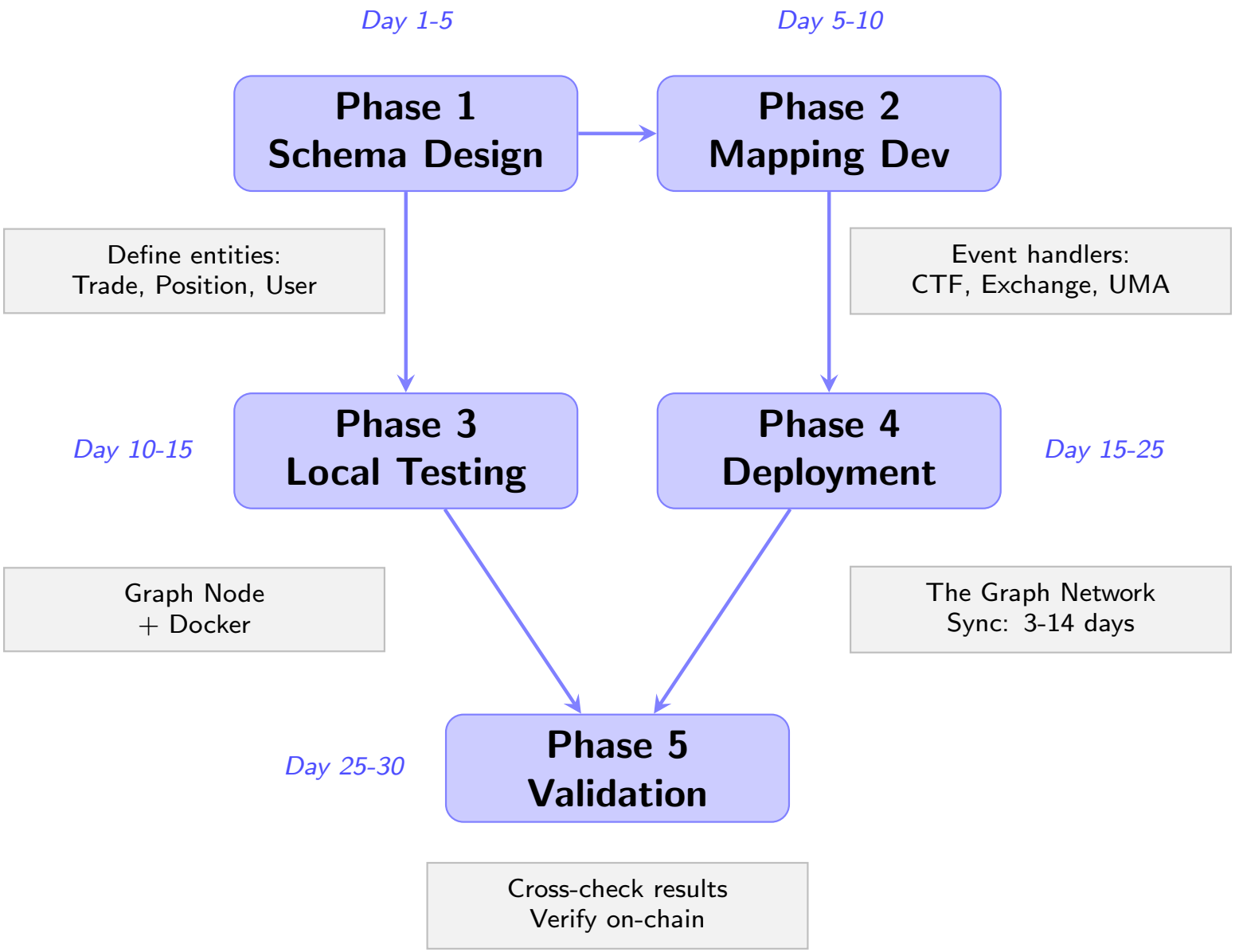
Why Custom Subgraph?

1. **Engineering Efficiency**
   - Automated indexing vs. manual JSON-RPC parsing
   - Built-in reorg handling and data consistency
   - GraphQL API eliminates complex data joins
2. **Development Velocity**
   - 10x faster queries than direct RPC calls
   - Declarative schema design with TypeScript
   - Hot-reload development environment
3. **Research Flexibility**
   - Custom aggregations for behavioral analysis
   - Iterative schema evolution without data loss
   - Deterministic and reproducible results

Technical Validation

Polymarket's production infrastructure uses The Graph - proven scalability for 50M+ blocks and billions of events

Implementation Plan



Expected Outcomes & Timeline

Deliverables

- **Complete Historical Dataset**
  - 50M+ blocks indexed
  - All trades since 2020
  - User wallet mappings
- **Analytics API**
  - GraphQL endpoint
  - Custom aggregations
  - Real-time updates
- **Research Insights**
  - User behavior patterns
  - Market efficiency metrics
  - Event impact analysis

Timeline

Week 1-2: Schema design & development

Week 3: Local testing & optimization

Week 4: Deploy to The Graph

Week 5-6: Historical sync (background)

Week 7: Validation & debugging

Week 8: Analysis queries ready