Building While Flying: Lessons Learned from Operating and Developing a Graph Service with Apache Tinkerpop
Who are we?

Keith Lohnes  
krlohnmes@us.ibm.com

David Pitera  
dpitera@us.ibm.com
What is a graph?
What is a graph?

Nodes

Edges
Traversing a graph
Traversing a graph
Traversing a graph
Graph Use Cases

- Recommendation engines
- Anomaly/Fraud Detection
- Network Analysis/Route planning
- Social Networks
- Identity/Access Management
- Graph-based search
Graphs are built for complex or highly networked data.

RDBMSs are designed for ordered data or data that can be easily structured.
Trees v. Graphs

Trees model order

Graphs model complexity
Property Graphs

[Diagram showing a property graph with nodes and edges labeled with properties such as name, age, and weight.]
Recap

- Graphs have nodes and edges
- Property graphs: nodes and edges have properties
- Edges are treated as first class citizens
- Queries can start at nodes or edges
- Traversals are fast across many graph elements
What does it take to turn an OSS project into a reliable and highly available database aaS offering?

- Architecture support for multi-tenancy
- Turning the awesome TP3 software into something that suits our needs
- Operations
IBM Graph Architecture

https://bluemix.net

IBM Graph Beta
Apache Tinkerpop
Node.js, Titan, Elasticsearch, and Cassandra
Cloud vs OSS On Premise
Multi-tenancy
Multi-tenancy

Multiple users operating inside a single environment
Multi-tenancy

Multiple users operating inside a single environment

Conquering hurdles:
Multi-tenancy

Multiple users operating inside a single environment

Conquering hurdles:
- Cassandra keyspace
Multi-tenancy

Multiple users operating inside a single environment

Conquering hurdles:

- Cassandra keyspace
- Groovy sandbox: blacklists, “def”
Multi-tenancy

Multiple users operating inside a single environment

Conquering hurdles:
- Cassandra keyspace
- Groovy sandbox: blacklists, “def”
- Forced indexing
#DevOps
Pull Request
Jenkins PR Test/Code Review
Merge to master
Jenkins build and test
Publish artifacts
Auto Deploy ZDD to Dev/QA
Live env testing
IBM Graph

Keith Lohnes  krlohes@us.ibm.com

David Pitera  dpitera@us.ibm.com

IBM Graph is an easy-to-use, fully-managed graph database service for storing and querying data points, their connections, and properties. IBM Graph offers an Apache TinkerPop3 compatible API and plugs into your Bluemix application seamlessly. This service can be used for building recommendation engines, analyzing social networks, and fraud detection.

- **Highly Available**
  Architected to ensure the service is always up and your data is always accessible.

- **Managed 24x7**
  Our experts monitor, manage & optimize everything in the stack 365 days a year.

- **Scale Seamlessly**
  Start small and scale on demand as your data grows.

- **Powered by Apache TinkerPop3**
  IBM Graph is based on the TinkerPop stack for building high-performance graph applications.

Pick a plan

<table>
<thead>
<tr>
<th>Plan</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅ Entry</td>
<td>1 dedicated graph per instance</td>
</tr>
</tbody>
</table>