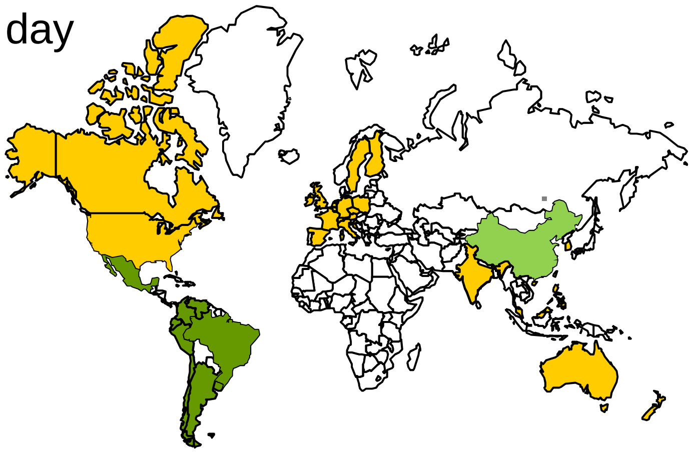
More Best Practices for Large-Scale Websites
Lessons from eBay

Randy Shoup
eBay Chief Engineer

QCon San Francisco
November 3, 2010

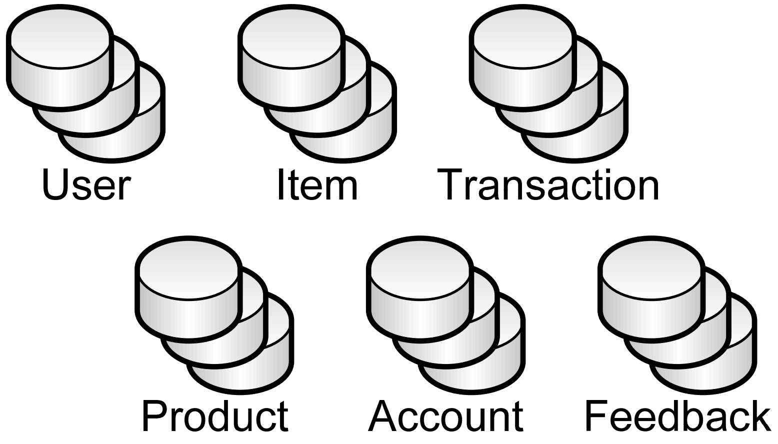
Challenges at Internet Scale

- eBay manages …
  - Over 90 million active users worldwide
  - Over 1 million mobile shoppers and >11 million app downloads
  - Over 220 million items for sale
  - Over 10 billion URL requests per day
- … in a dynamic environment
  - Tens of new features each week
  - Roughly 10% of items are listed or ended every day
- … worldwide
  - In 39 countries and 10 languages
  - 24x7x365
- … using
  - >10K Java application servers
  - >5K search engine nodes
  - >1K database instances
  - >85 billion read / write operations per day

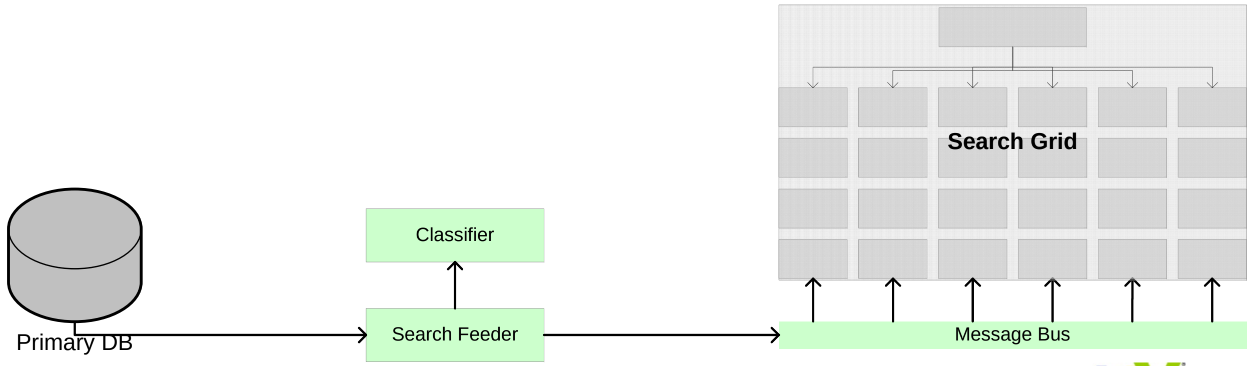


Architectural Lessons (round 1)

- **1. Partition Everything**
  - Functional partitioning for processing and data
  - Horizontal partitioning for data (“*shards*”)

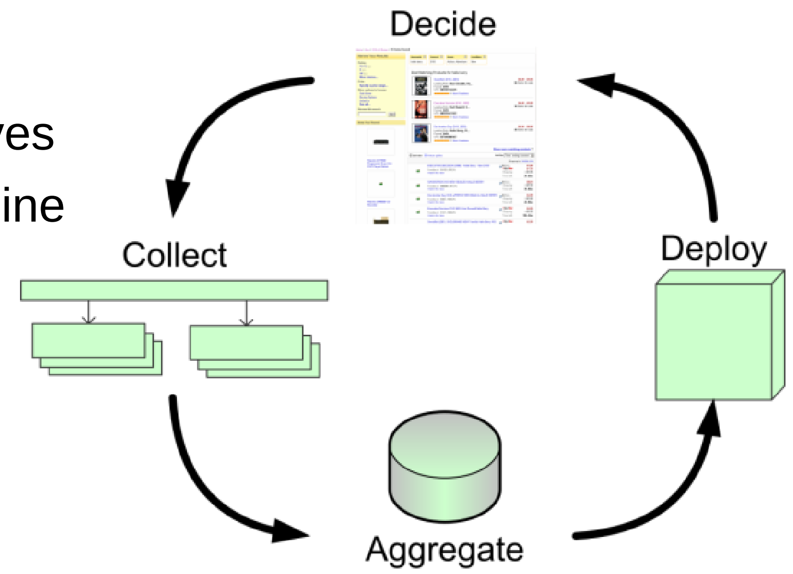


- **2. Asynchrony Everywhere**
  - Event-driven queues and pipelines
  - Multicast messaging
  - Batch processing

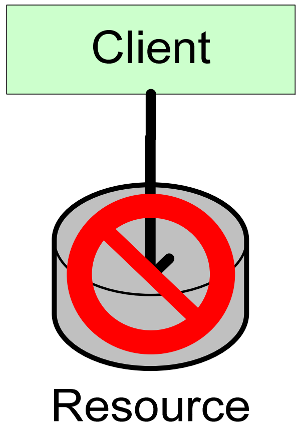


Architectural Lessons (round 1)

- **3. Automate Everything**
  - Components adaptively configure themselves
  - Heavily leverage feedback loops and machine learning

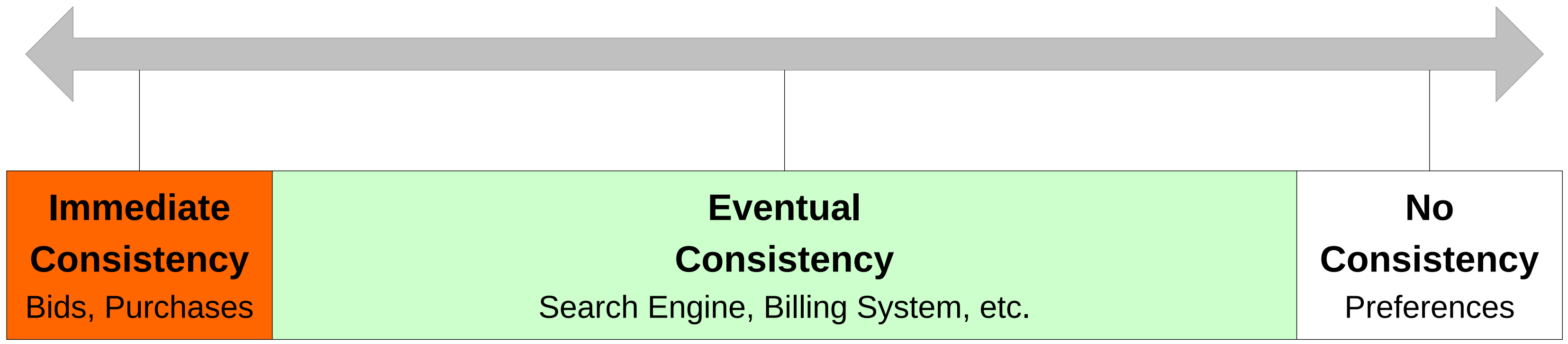


- **4. Everything Fails**
  - Extensive monitoring for rapid failure detection
  - Timeouts and retries part of every operation
  - Gracefully degrade functionality under failure
  - *Failure cases are normal, not exceptional*



Architectural Lessons (round 1)

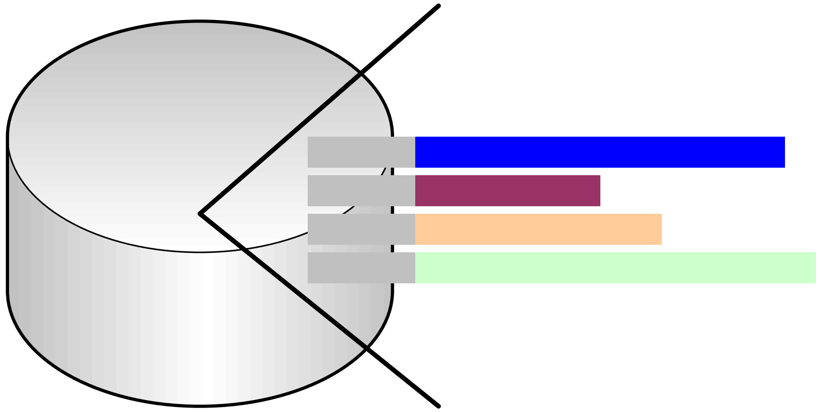
- **5. Embrace Inconsistency**
  - Consistency is a spectrum, and managed (in)consistency is the rule
  - *No distributed transactions (!)*
  - Minimize inconsistency through state machines and careful ordering of operations
  - Eventual consistency through asynchronous recovery or reconciliation



Lesson 6: Expect (R)evolution

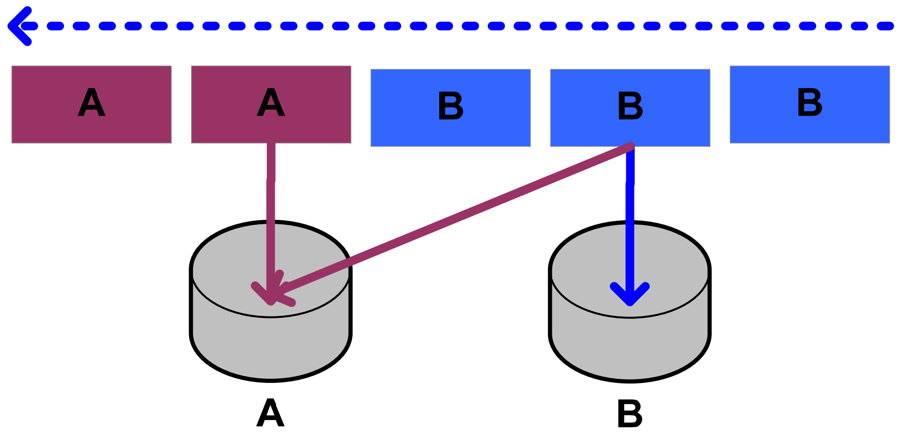
- **Change is the Only Constant**
  - New entities and data elements
  - Constant infrastructure evolution
  - Regular data repartitioning and service migration
  - Periodic large-scale architectural revolutions

- **Design for Extensibility**
  - Extensible Data: Flexible schemas
    - Extensible interfaces (e.g., key-value pairs)
    - Heterogeneous object storage
  - Extensible Processing: Events
    - Between systems, communicate via events
    - Within system, control processing pipeline via configuration



Lesson 6: Expect (R)evolution

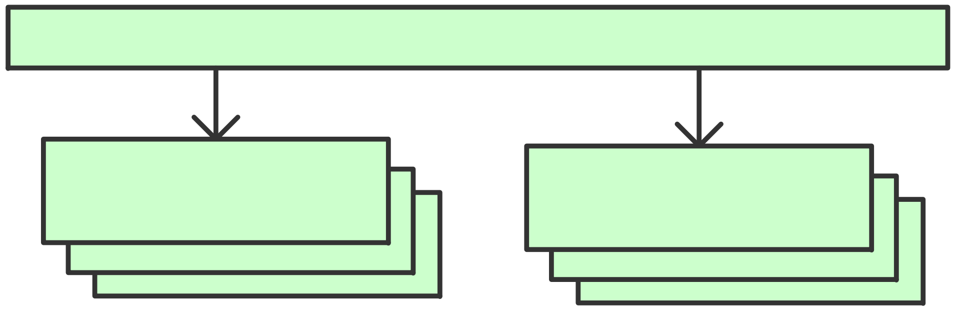
- **Incremental System Change**
  - Decompose every system change into incremental steps
  - Every step maintains strict forward / backward compatibility for data and interfaces
  - Multiple versions and systems coexist constantly
    - Every change is a rolling upgrade
    - *Transitional states are normal, not exceptional*
    - Version A -> A|B -> B|A -> Version B
  - Dual data processing and storage ("dual writes")



Lesson 7: Dependencies Matter

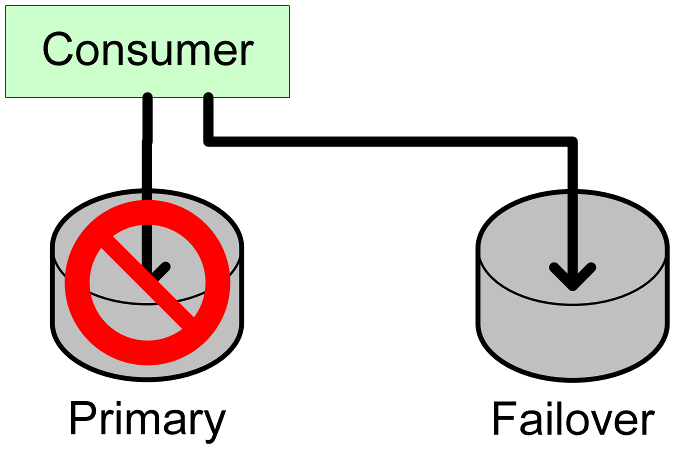
- **Carefully Manage Dependencies**
  - Minimize dependencies between systems
  - Depend only on abstract interface and virtualized endpoint
  - Quality-of-service guarantees must be explicit (expected latency, throughput)

- **Monitor Dependencies Ruthlessly**
  - Real-time dependency monitoring is essential for problem diagnosis and capacity provisioning
  - Registries say What It Should Be (WISB) but only monitoring provides What It Really Is (WIRI)



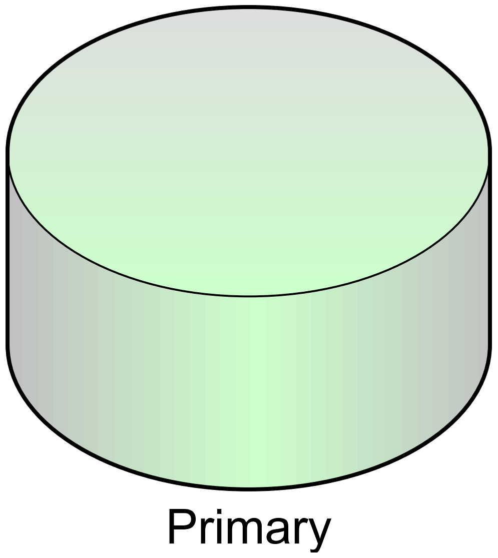
Lesson 7: Dependencies Matter

- **Consumer Responsibility**
  - It is fundamentally the consumer’s responsibility to manage unavailability and quality-of-service violations
  - (Un)availability is an inherently *Leaky Abstraction*
    - 1st Fallacy of Distributed Computing: “The network is reliable”
  - Recovery is typically use-case-specific
    - How critical is the operation? How strong is the dependency?
  - Standardized dependency-management patterns can help
    - Sync or async failover, degraded function, sync or async error, etc.



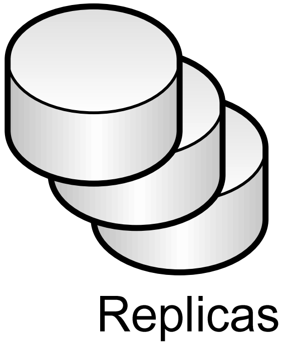
Lesson 8: Respect Authority

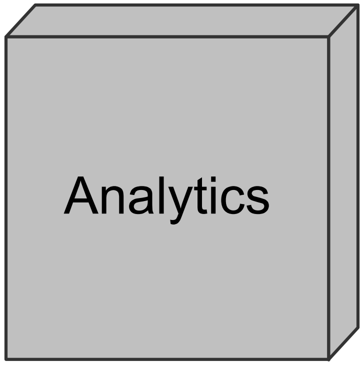
- **Authoritative Source ("System of Record")**
  - At any given time, every piece of (mission-critical) data has a single logical System of Record
  - Authority can be explicitly transferred (failure, migration)
  - Typically transactional system

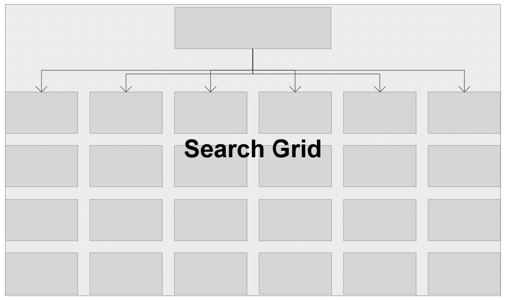


Lesson 8: Respect Authority

- **Non-authoritative Sources**
  - Every other copy is derived / cached / replicated from System of Record
    - Remote disaster replicas
    - Search engine
    - Analytics
    - Secondary keys
  - Relaxed consistency guarantees with respect to System of Record
  - Optimized for alternate access paths or quality-of-service properties
  - Perfectly acceptable for most use-cases

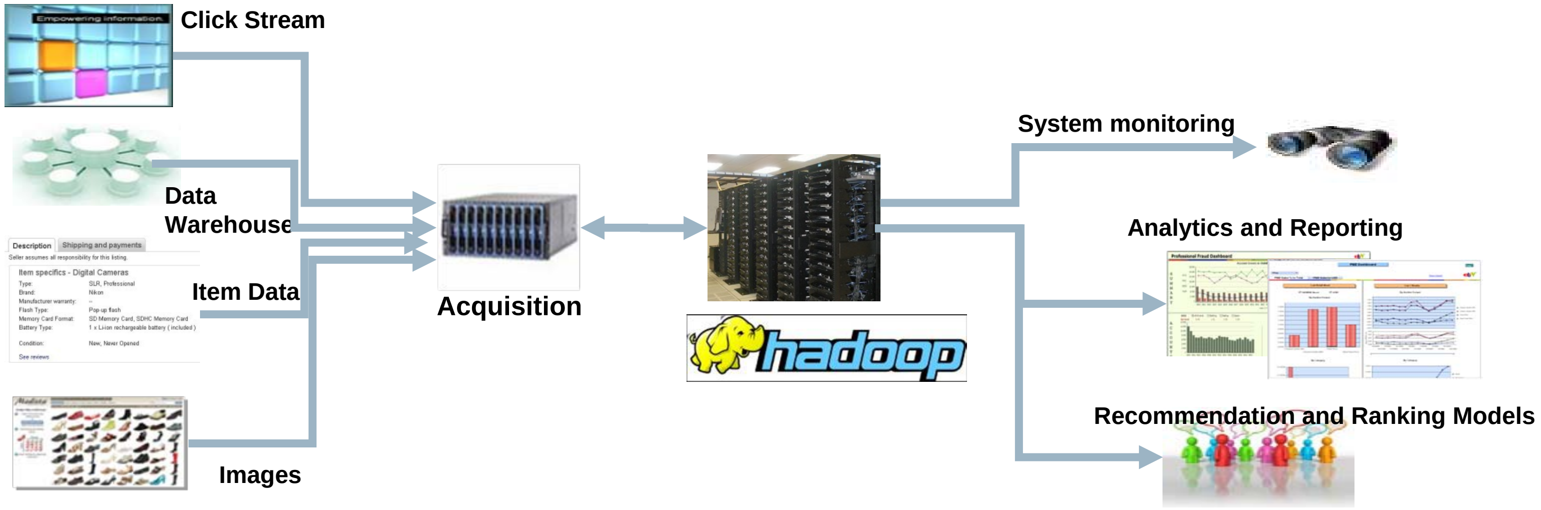






Lesson 9: Never Enough Data

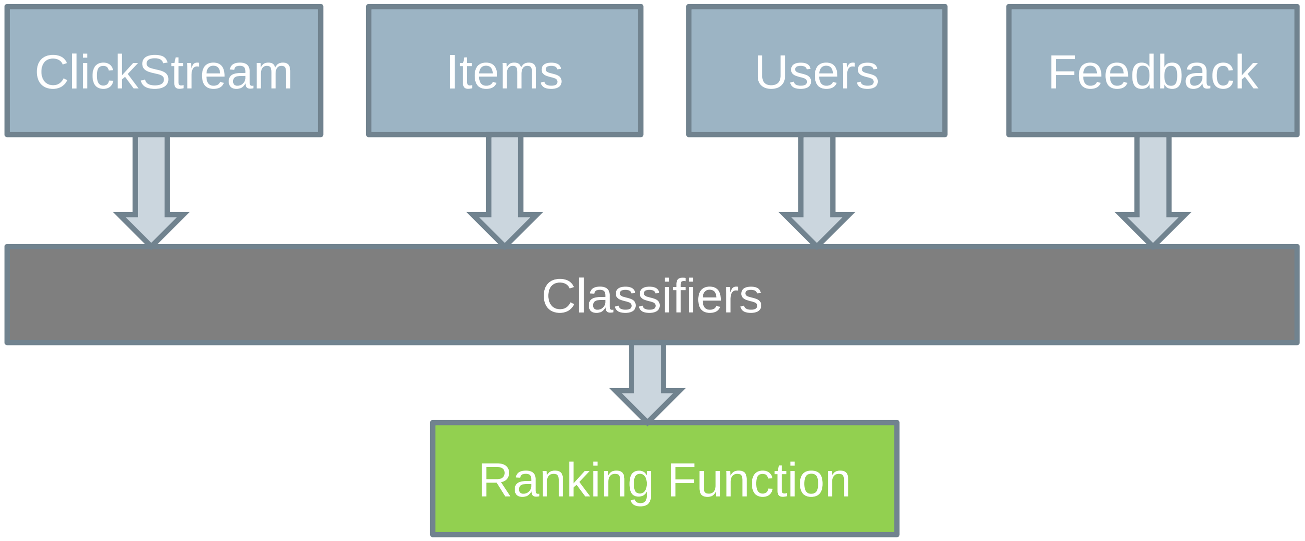
- **Collect Everything**
  - eBay processes 50TB of new, incremental data per day
  - eBay analyzes 50PB of data per day
  - Every historical item and purchase is online or nearline



eBay has the world's largest Teradata warehouse and is building one of the world's largest Hadoop clusters

Lesson 9: Never Enough Data

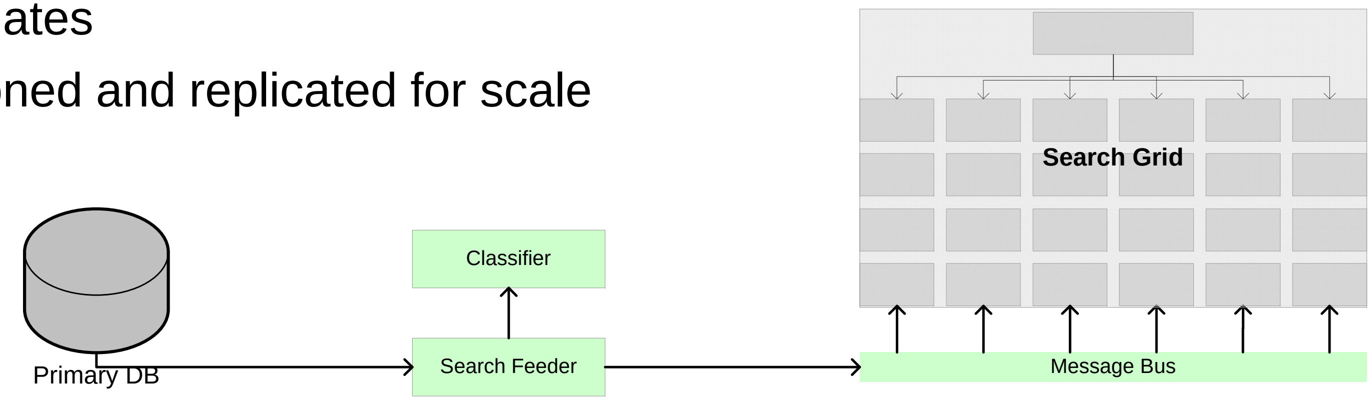
- **Example: Machine-Learned Search Ranking**
  - Learn models and ranking functions based on thousands of factors
    - User behavior, query factors, item factors, seller factors, etc.
  - Drive purchase recommendations



Predictions in the long tail require massive data

Lesson 10: Custom Infrastructure

- **Right Tool for the Right Job**
  - Need to maximize utilization of every resource
    - Data (memory), processing (CPU), clock time (latency), power (!)
  - One size rarely fits all, particularly at scale
- **Example: Real-Time Search Engine**
  - In-memory inverted index with real-time lock-free index updates
  - Real-time feeder pipeline with multicast messaging to the search engine
  - Highly flexible schema with arbitrary ranking functions, expressions, and aggregates
  - Partitioned and replicated for scale



400 million queries and 500 million updates per day

Ten Lessons

1. Partition Everything
2. Asynchrony Everywhere
3. Automate Everything
4. Everything Fails
5. Embrace Inconsistency
6. Expect (R)evolution
7. Dependencies Matter
8. Respect Authority
9. Never Enough Data
10. Custom Infrastructure

Thank you!

… eBay is Hiring …

http://ebaycareers.com

- Mobile Applications
- Presentation Engineering
- Distributed Systems
- Automation
- Information Retrieval
- Machine Learning
- …