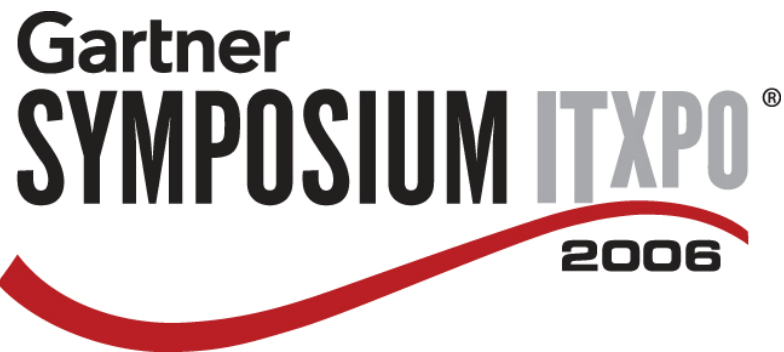


# The Death of the Database



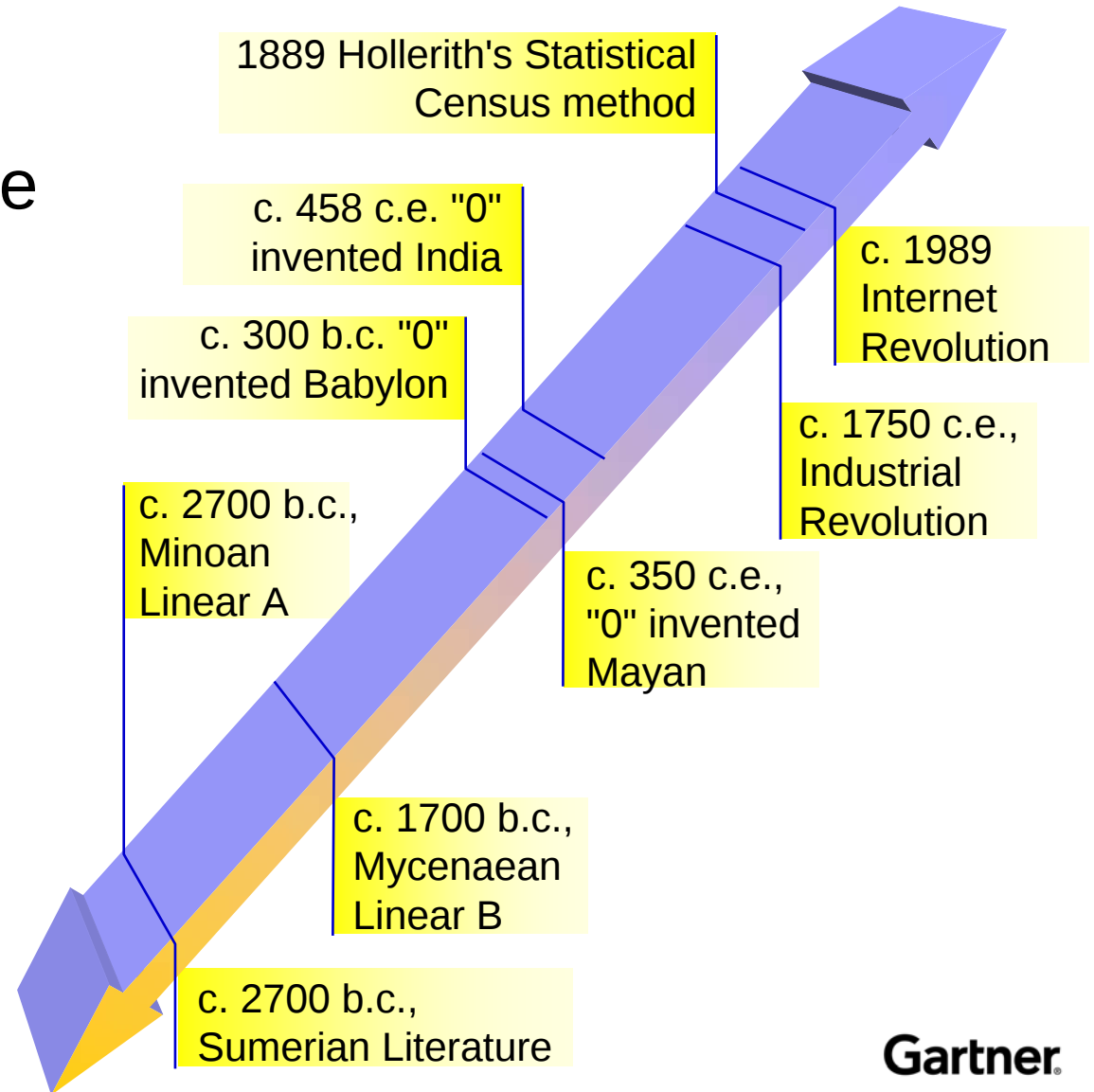
Mark Beyer

Notes accompany this presentation. Please select Notes Page view.  
These materials can be reproduced only with Gartner's official approval. Such approvals may be requested via e-mail — [vendor.relations@gartner.com](mailto:vendor.relations@gartner.com).

**Gartner®**

# Database-Dependent SQL Will Be Taken Away and Context Will Be Real Time

- Defining persistence
- Think of verbs
  - Storage
  - Access
  - Delivery
  - Utilization/usage



# Key Issues

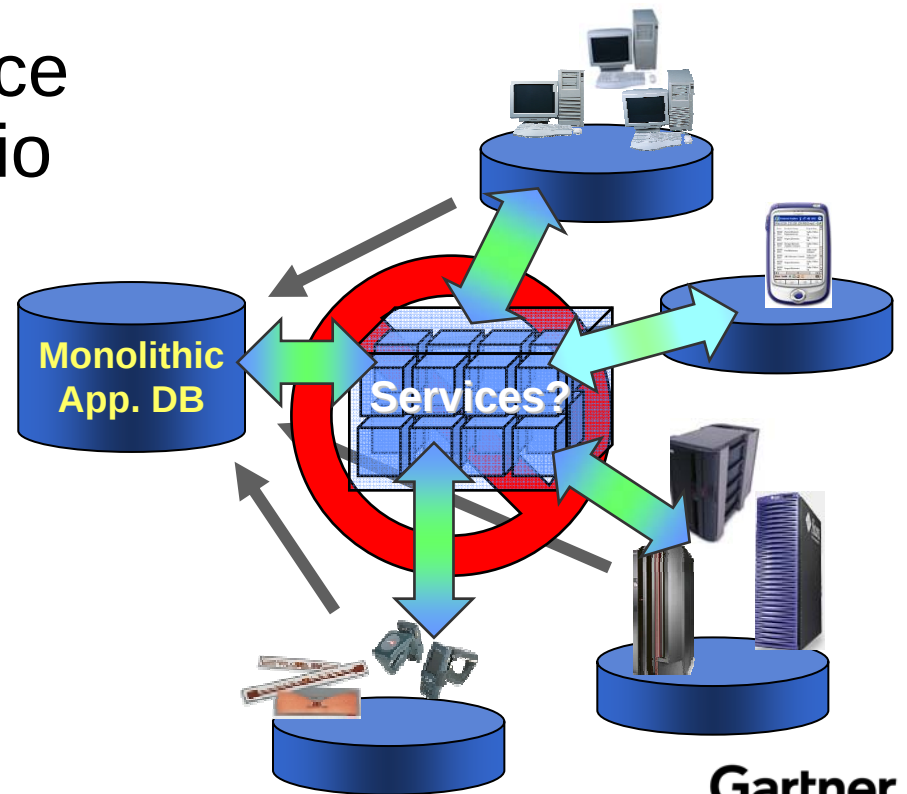
- 1. How quickly will databases become obsolete?**
- 2. Why do service-oriented architectures change data persistence beyond recognition?**
- 3. Will any of your products and skills remain relevant?**

# Key Issues

- 1. How quickly will databases become obsolete?**
2. Why do service-oriented architectures change data persistence beyond recognition?
3. Will any of your products and skills remain relevant?

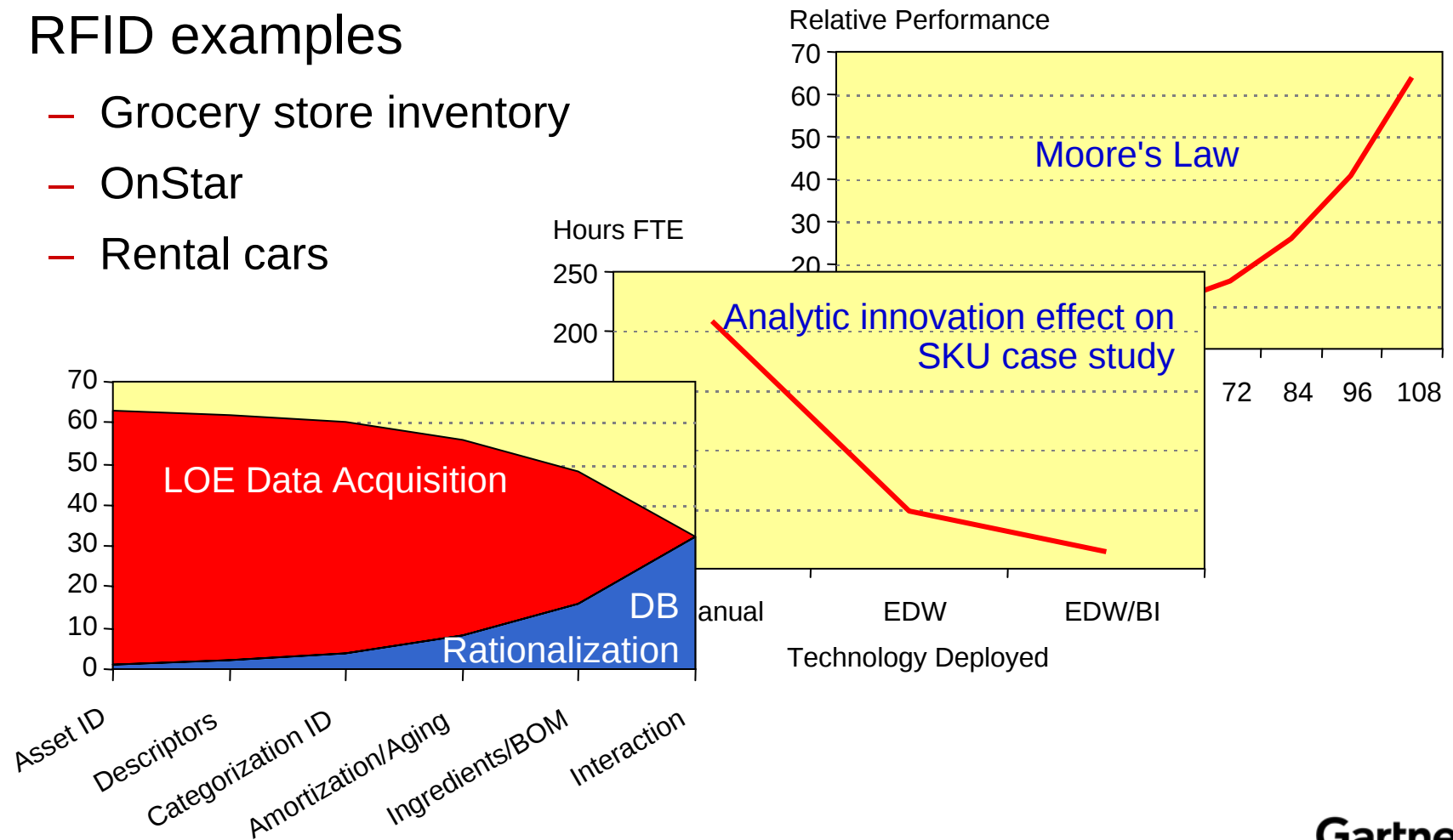
# Storage Changes

- Grid — the network becomes the database?
- Queues
- Programmatic persistence yields to a policy scenario
- Indexing beyond the database
- Microscale persistence at the asset



# Storage Changes: Distribution of Persistence

- Everything carries its own data
- RFID examples
  - Grocery store inventory
  - OnStar
  - Rental cars



# Key Issues

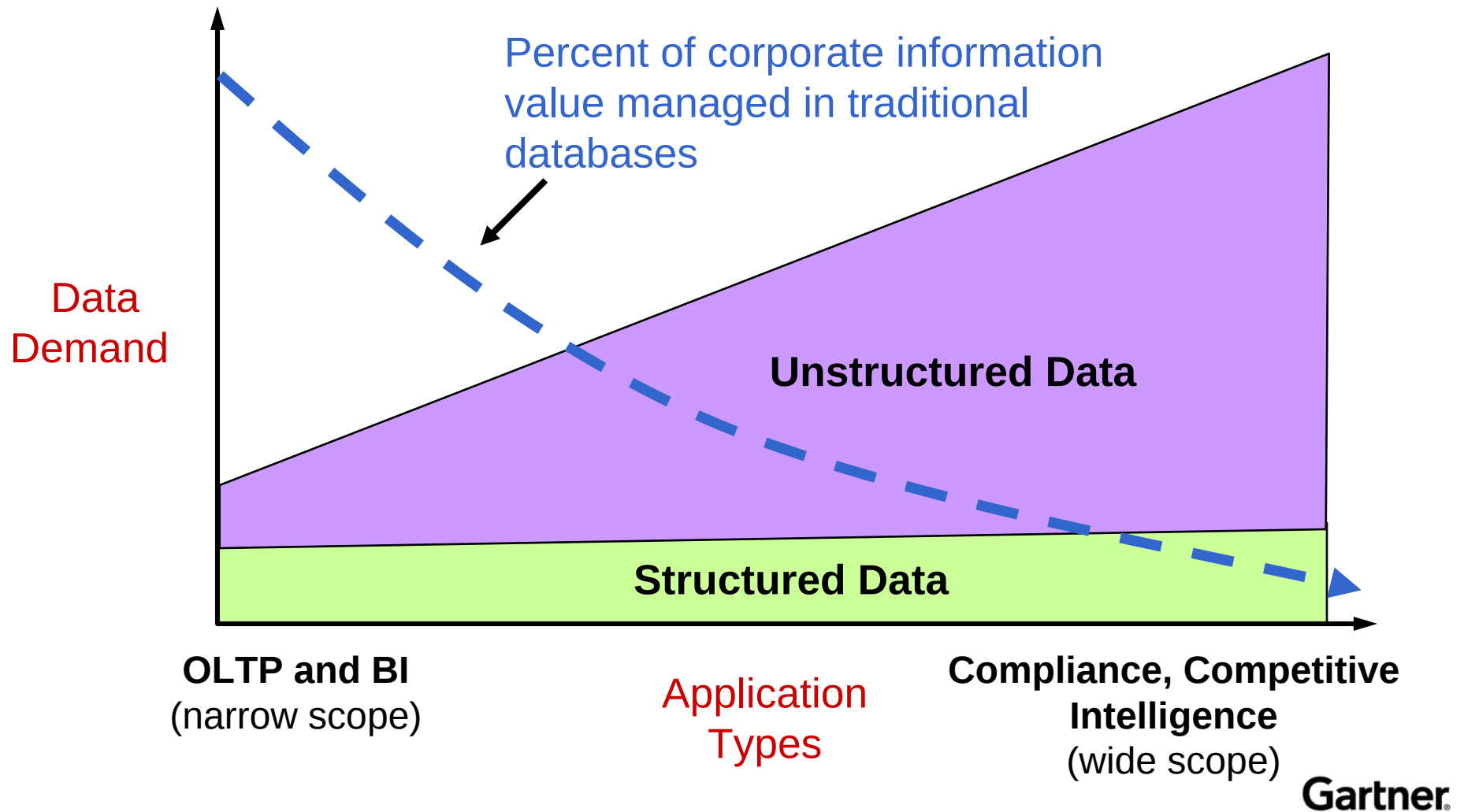
1. How quickly will databases become obsolete?
2. Why do service-oriented architectures change data persistence beyond recognition?
3. Will any of your products and skills remain relevant?

# Access Changes: Emerging Access Mechanisms

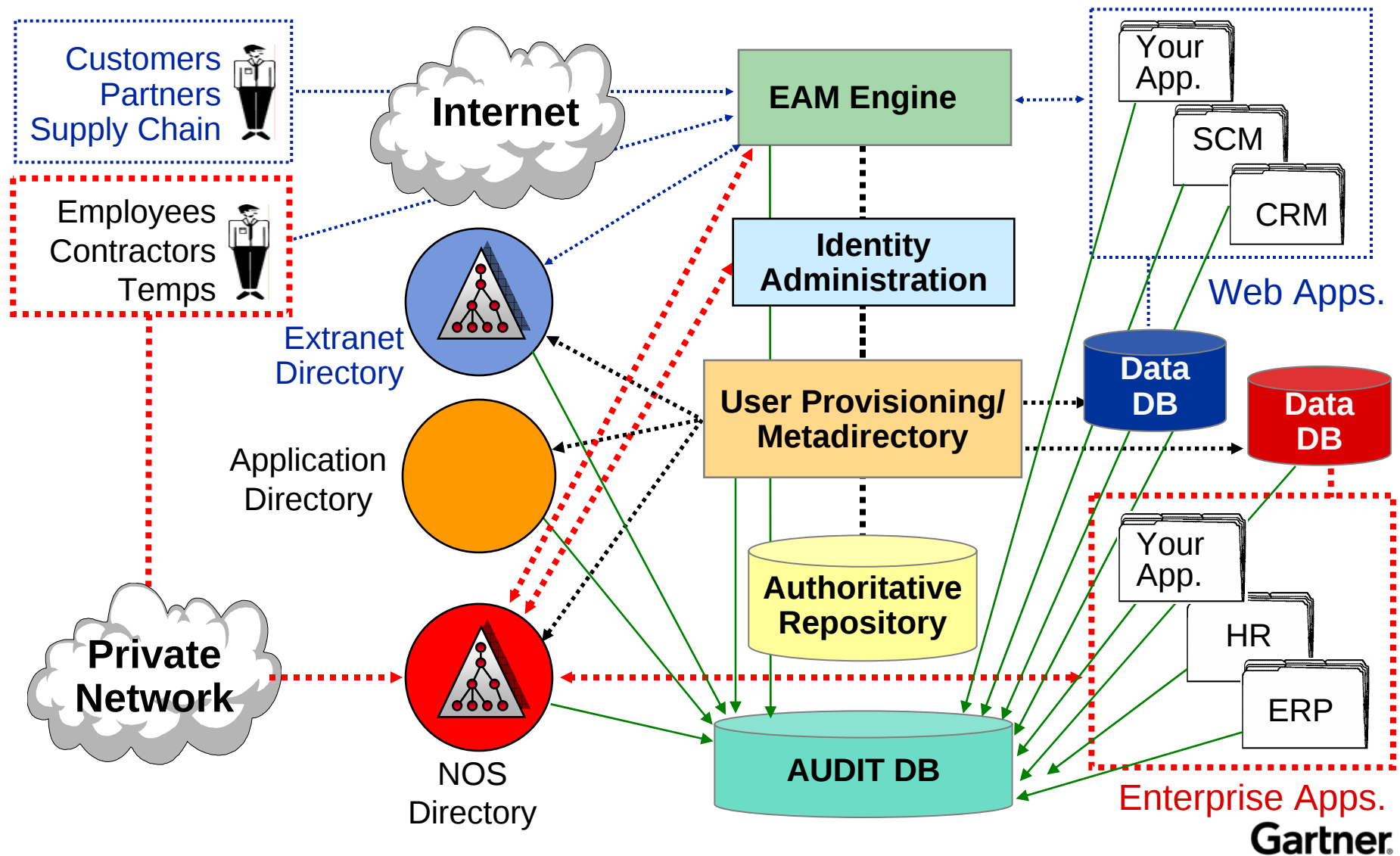
- Traditional access mechanisms included requirements such as location and method
  - Data of interest in DBMSs
  - Standardized access methods
- Emerging access mechanisms are designed around reading data content
  - XQuery
  - Search data need not be centralized
- Hypertext as an engine of state?

Develop new policies for persistence based on mechanisms other than DBMS

# Access Changes: Center of Gravity Moves Toward Unstructured Data

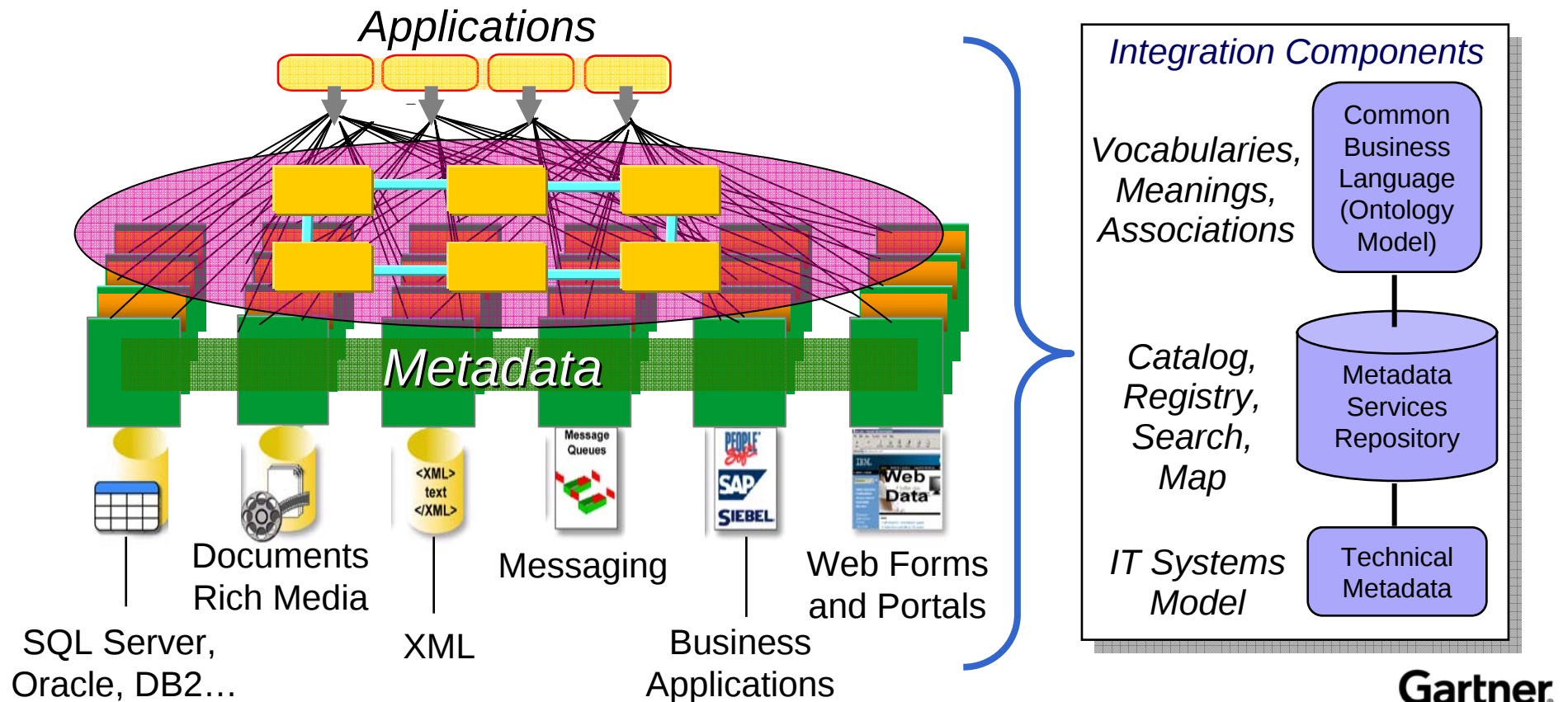


# Access Changes: Security Rises Above the Database



# Delivery Changes: Metadata Breaks the Binding

- Dynamic metadata repositories
- Dynamic decision by applications
- Applications obtain data "addresses" from services



# Delivery Changes: Off the Beaten Path, Old Compromises Are Gone

## ■ SOA

- Implied universal connectivity
- No difference in access/delivery of persistent and ephemeral information
- Means that persistence can become implicit and policy-based

## ■ Everything looks ephemeral

- Mechanisms federate highly redundant, overlapped and distributed data for reference information
- Transactional information flow is controlled by process engines that support persistence policy at the macro level
- Micro-level policies are implemented in middleware (for example, queues backed by persistence storage, write through distributed caches)

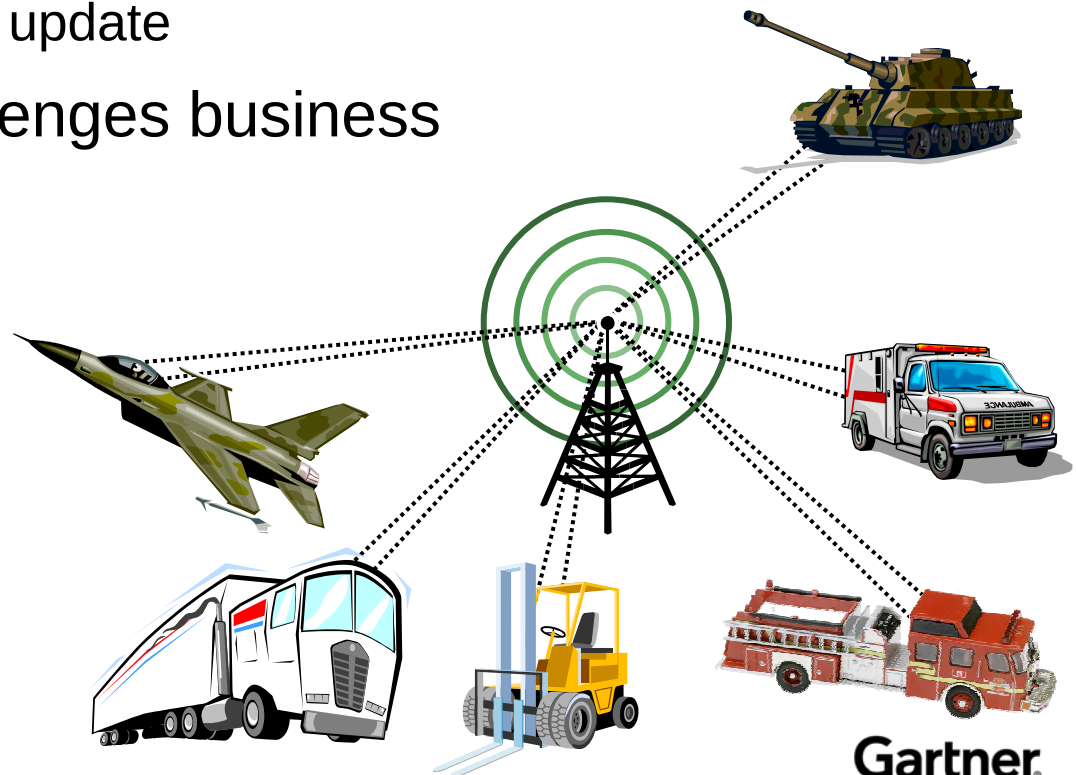
# Key Issues

1. How quickly will databases become obsolete?
2. Why do service-oriented architectures change data persistence beyond recognition?
3. Will any of your products and skills remain relevant?

# Usage Changes: Active Decision Processing

- Operational applications will actively locate resources
  - Range, asset profiles, operational intensity
  - Data rows do not have to be placed into a static repository for append or update
- Business function challenges business process-based design

Identify vendors  
concentrating on  
the services and  
policy vision



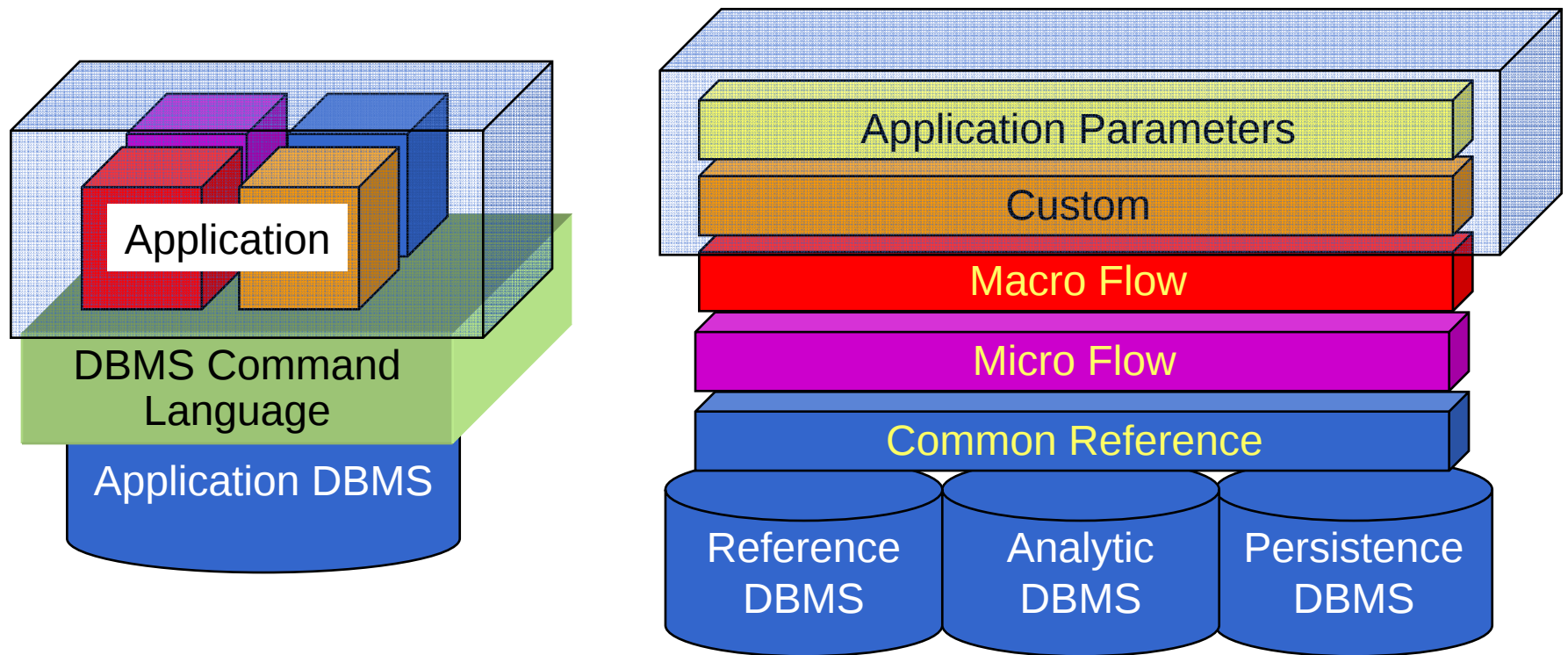
# Usage Changes: Documentation and Reporting

- Documentation moves to an active state
  - Warning labels
  - Ingredients/BOM
- Reporting becomes an archive activity
  - Actual data state vs. interpreted data states
- Insurance claim status, for example
  - Workflows transformed from database tracking to a distributed process engine
  - Process "state" is examined "in flight" instead of a representation in a database



# Data Persistence and Policy Layers

- Default standard is "static resume" for all activity and controlled by each application using DBMS commands
- Persistence becomes a series of reusable policies only, invoking persistence when the application SLA demands it



# Usage Changes: Policy Enables the Distribution of Data Persistence

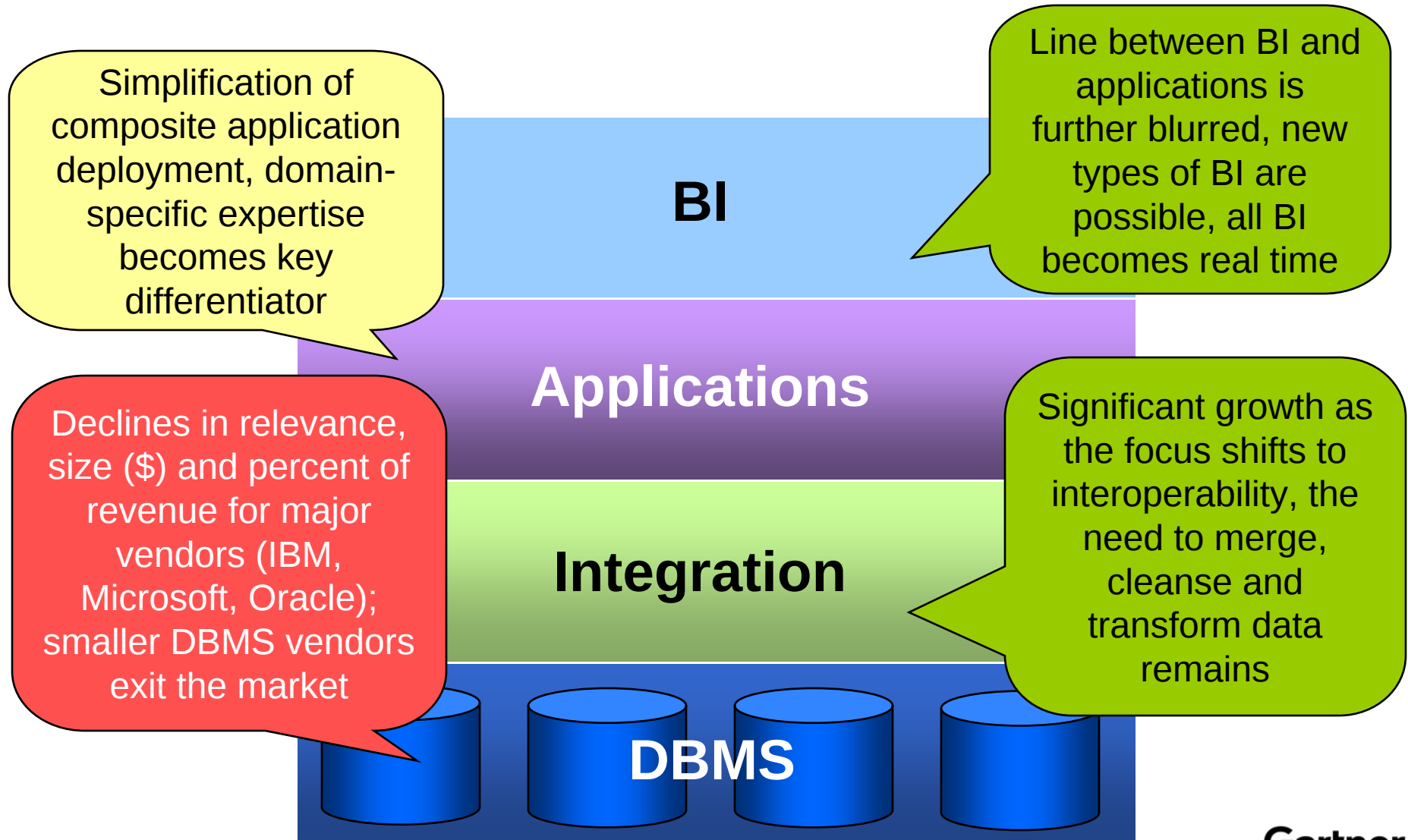
- Process tracking data disappears
- Data points to support rules rationalization become the dominant data type
  - Situational analysis
  - Definitional parameters
- Reporting and analysis dichotomy is more precise
  - Historic vs. operational reporting
  - Historic vs. operational analysis
- Policy-based persistence

Primary database deployment will shift to master data, metadata and historic data repositories by 2011 (0.6 probability).

# Persistence Becomes a Policy

- Persistence becomes an attribute of all parts of the system, not just concentrated in the database
- Policy-based persistence is the norm, and mechanisms and architecture reflect the various policy scopes
  - Classic "database" fits only the historic scope
- Federation succeeds over limited domains
  - Continues to be challenged by the "flood" of more data used more often

# Market Impact: Winners, Losers and Pervasive Change



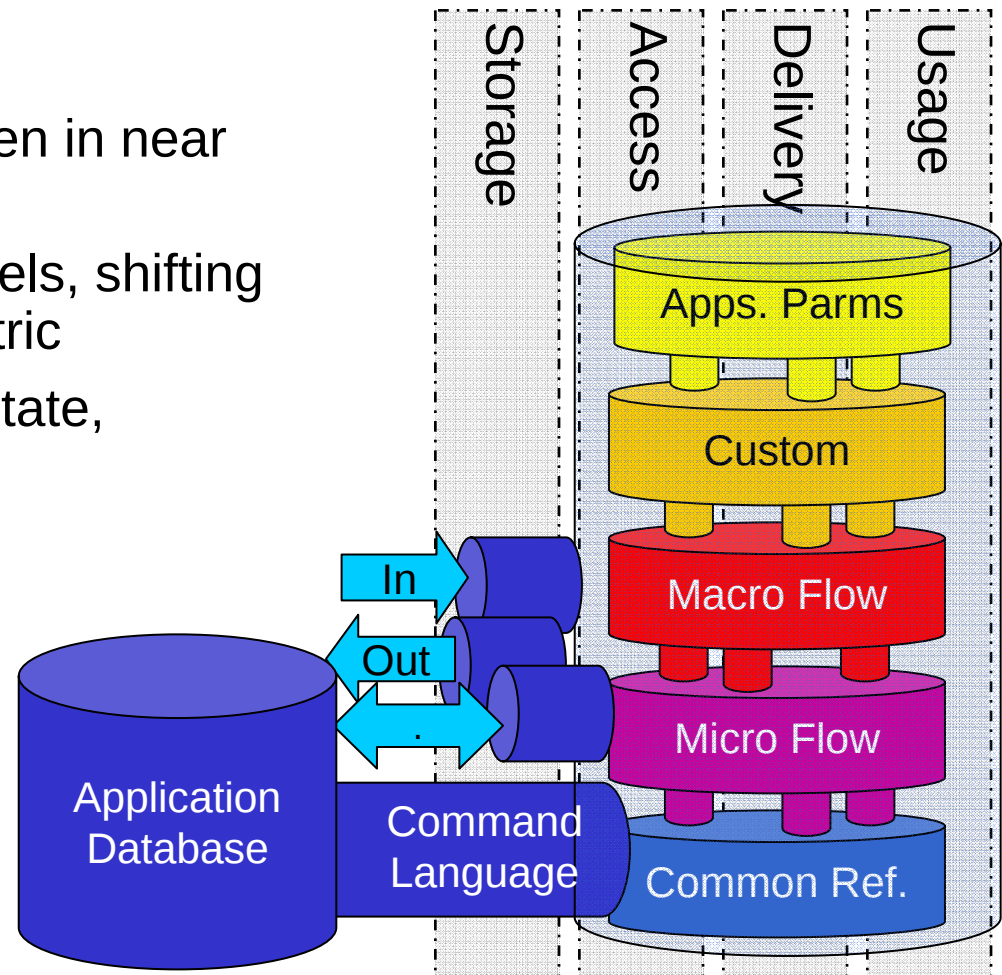
# Maintenance Fades: Utilization Knowledge Is Dominant

- Current DBA skills become less relevant
  - Architectural skills for information representation remain valid
  - Technology skills will decline in value (e.g., DBMS-specific knowledge)
- By 2009, the DBA will split into two roles (0.8 probability)
  - Repository administrator — catalog of sources, data-location expert including layout and model
  - Data service administrators — knowledgeable of statistics gathered, system reliabilities, data reliabilities, physical and logical data models



# Business Is Function, Not Sequence

- Speed of process execution
  - Demands data access happen in near real time
- Process-centric automation models, shifting from task-centric to process-centric
  - Broader representations of state, transferred less often
- Distribution of information: Information is highly distributed
  - Mechanisms that can treat the distributed engine as an entity for data management



# Recommendations

## ■ User Organizations

- Develop new applications for DBMS independence
- Develop new policies for persistence based on other mechanisms besides DBMS
- Create clear service levels for persistence of information when designing systems
- Foster overlap between middleware and DBA skills
- Identify vendors concentrating on the services and policy vision

## ■ Vendors

- Eliminate database command language dependencies in applications
- Acquire or partner with domain specialists
- Become first or best in a targeted domain of data management
- Enable interoperability with other domain specialists