

Making Leaders Successful Every Day

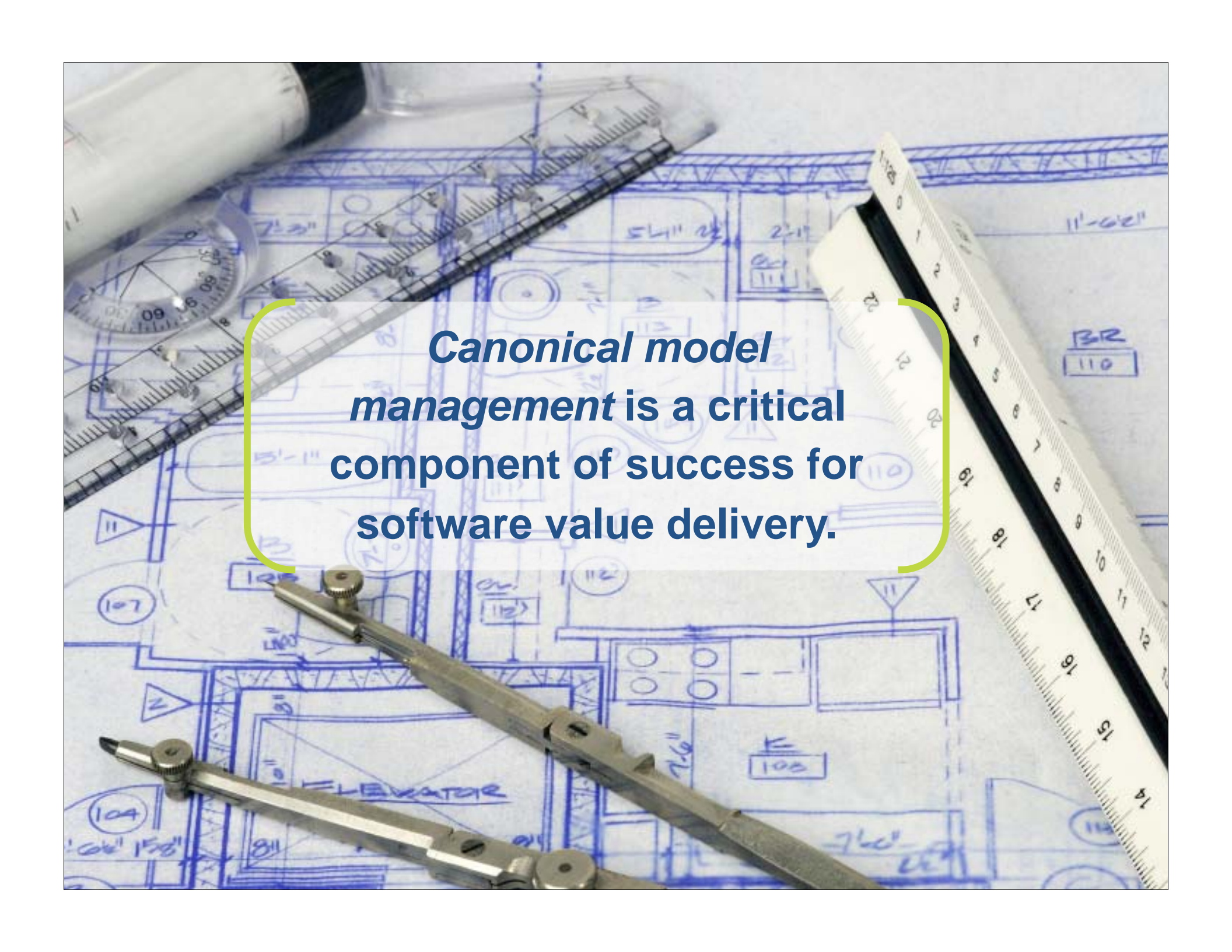
FORRESTER®

Canonical Model Management Best Practices And Key Trends

Mike Gilpin, VP / Research Director

May 9, 2011

FORRESTER®

The background of the image is a detailed architectural blueprint drawn in blue ink on white paper. The blueprint shows a complex floor plan with various rooms, corridors, and structural elements. Dimensions are written throughout, such as '5'-11"', '2'-11"', '11'-0"', '7'-3"', '5'-11"', '7'-0"', '10'-11"', '8'-11"', '7'-6"', '10'-0"', '15'-0"', and '8'-11"'. There are also labels for rooms, including 'ELEVATOR', '104', '107', '108', '110', '112', and '115'. A white ruler is positioned diagonally across the top right, and a pair of metal compasses is in the bottom center. A white marker is visible in the top left corner. The text 'Canonical model management is a critical component of success for software value delivery.' is overlaid in the center in a bold, dark blue font, enclosed in a light green rounded rectangular border.

***Canonical model management* is a critical component of success for software value delivery.**

Definition

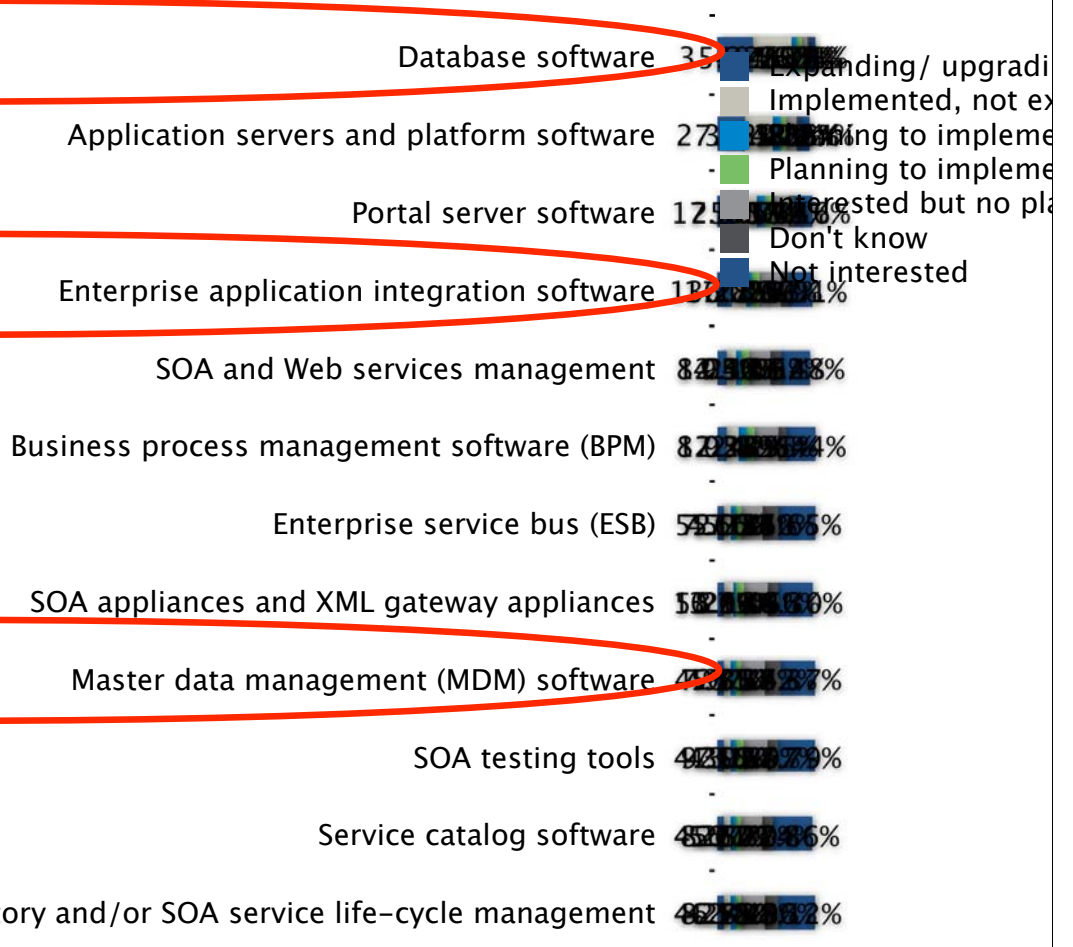
- A canonical information model is a model of the semantics and structure of information that adheres to a set of rules agreed upon within a defined context for *communicating* among a set of applications or parties.

Canonical models are more often about ***data in motion*** than *data at rest* – but not always

Canonical models may be about *more* than information

- § Information is just one of the “four elements” – also including services, processes, and events
- § Canonical services, processes, and events are also possible
- § So... the full “canonical model” *can* model all the semantics of business
- § But... such an effort may be too large in scope to be pragmatic

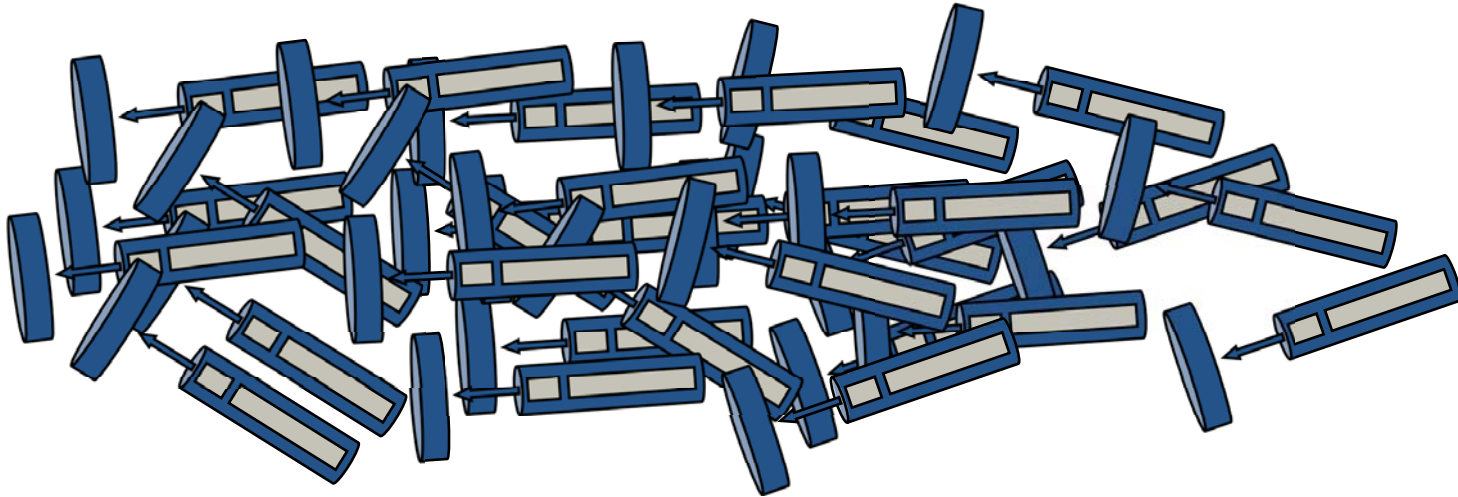
But we see less demand for MDM, which is related



Base: 934 North American and European software decision-makers

Source: Enterprise And SMB Software Survey, North America And Europe, Q4 2010

When services proliferate, model chaos reigns



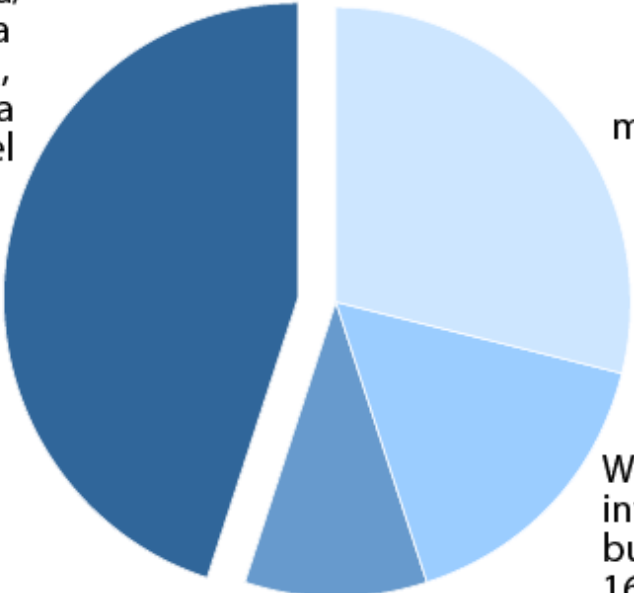
- § It gets too hard to integrate new applications into processes
- § Chaotic metadata also exacerbates data quality issues
- § Bottom line: *A chaotic information model reduces the value of information*

*But organizations often lack the **processes** and **technology** to avoid information model and service metadata chaos*

Canonical modeling is increasing

“Is your organization working on establishing a canonical information model as part of your SOA, data/information services, or data integration implementation?”

We are pursuing an SOA, data/information services, or data integration implementation, but we are not establishing a canonical information model
45%



We are working on a canonical information model at the enterprise level
29%

We are working on a canonical information model at the business unit level or lower
16%

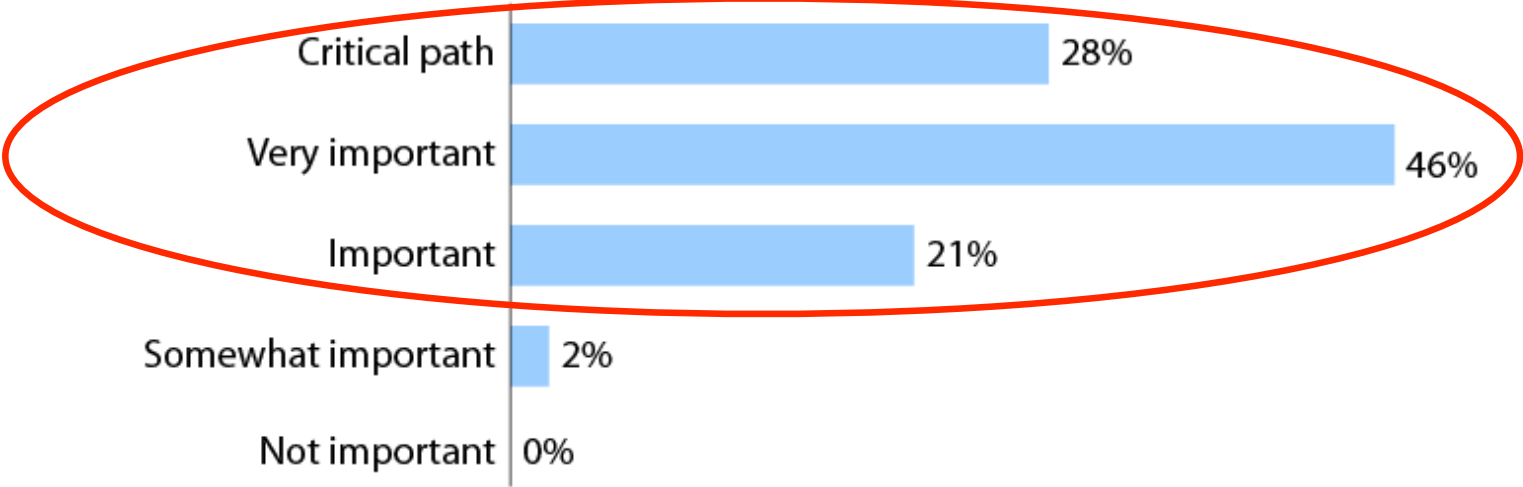
We are working on a canonical information model at the ecosystem level
10%

Base: 107 EA professionals

Source: November 2009 Global Information Architecture Online Survey

Data governance is crucial

“How important do you/did you rate data governance in the success implementation and rollout of your information architecture?”

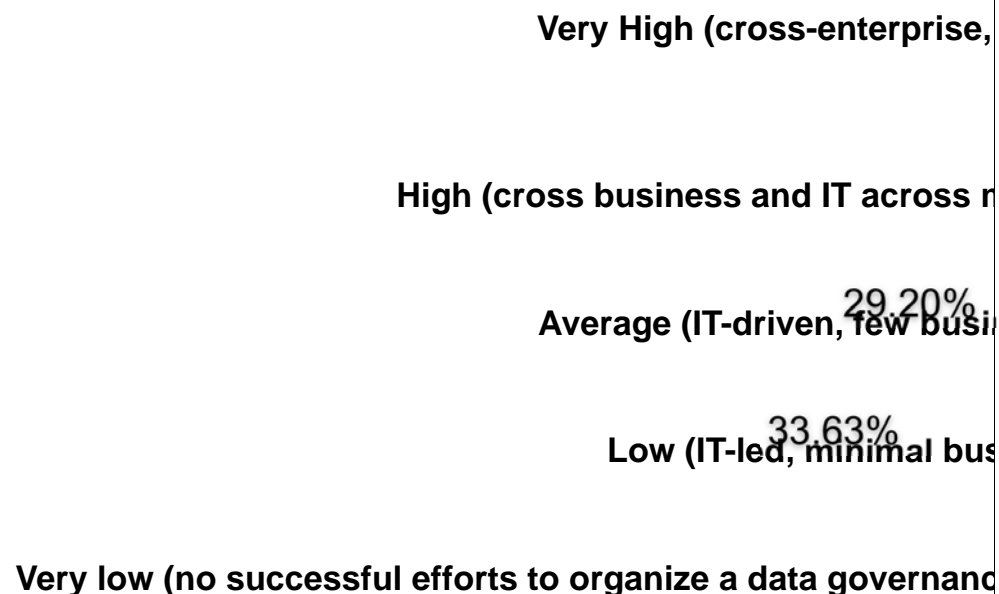


Base: 125 EA professionals
(percentages do not total 100 because of rounding)

Source: November 2009 Global Information Architecture Online Survey

Yet most organizations have immature data governance

“How would you rate your organization’s current level of data governance* maturity?”



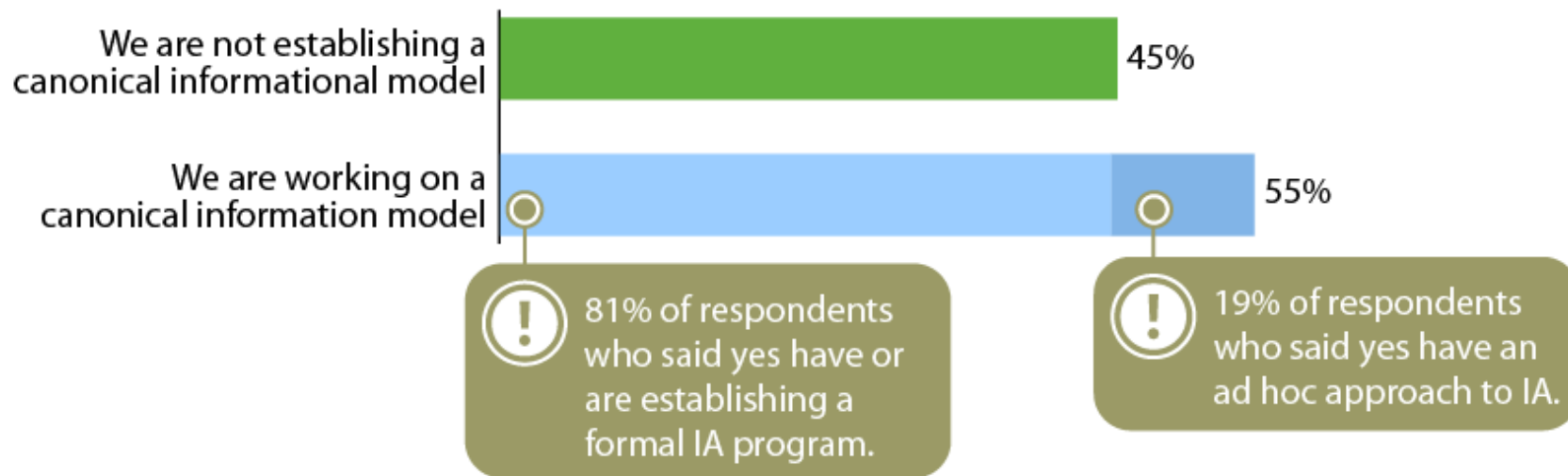
Base: 113 MDM- and/or data-quality-savvy IT professionals

* Data governance: the process by which an organization formalizes the fiduciary duty for the management of data assets critical to its success

Source: August 2009 Global Master Data Management/Data Quality Online Survey

Canonical modeling is part of information architecture

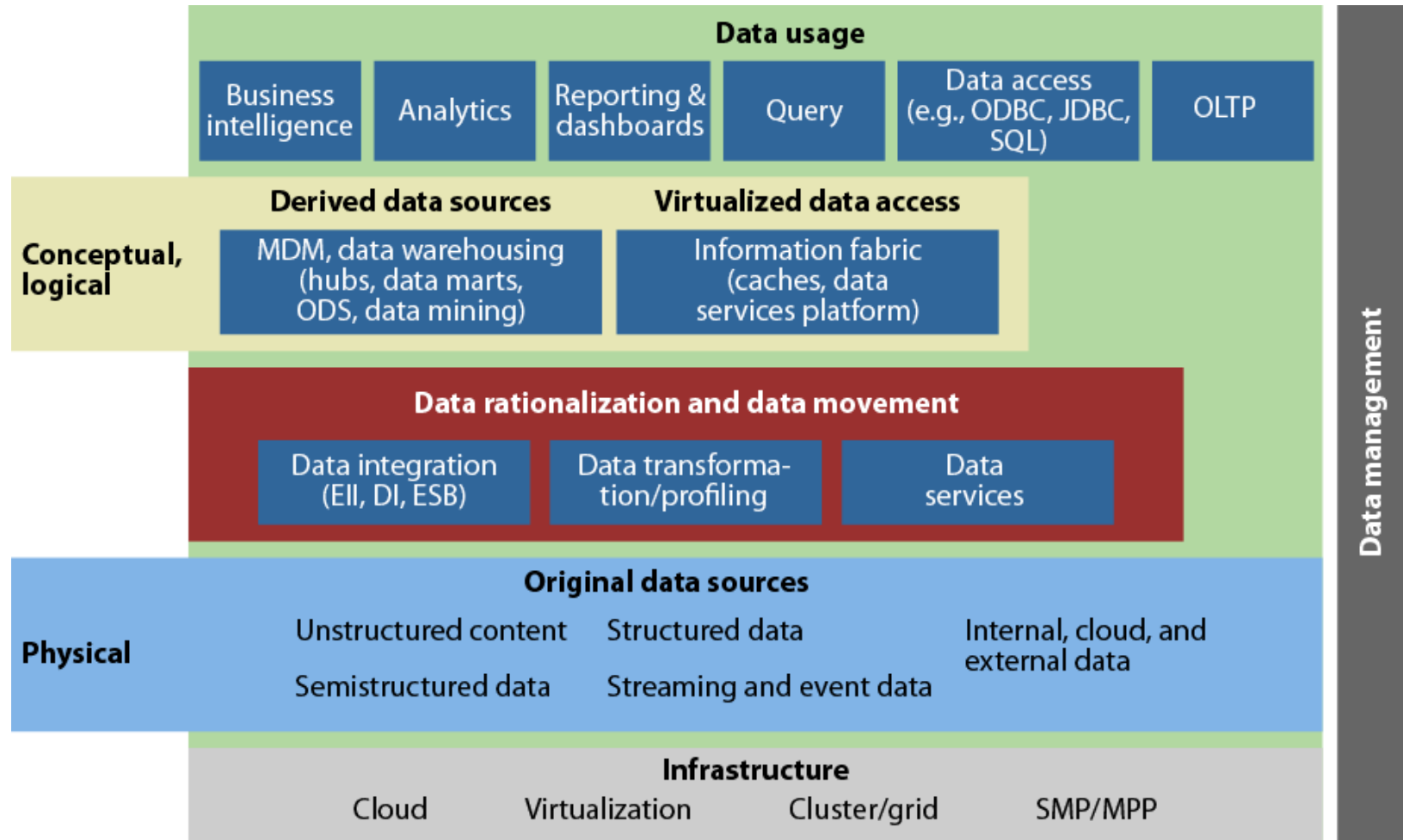
“Is your organization working on establishing a canonical information model as part of your SOA, data/information services, or data integration implementation?”



Base: 107 IT professionals pursuing SOA, data/information services, or data integration
("Don't know" responses not included)

Source: November 2009 Global Information Architecture Online Survey

Forrester's Data Management Reference Architecture



Lessons from recent case studies

- § Focus on a *limited* number of process domains at first, add incrementally
- § Industry, *application*, or other domain/ecosystem standards are crucial to success – NIEM, UCore, FIX, SWIFT, FpML, SEPA, ISO 20022, , Deriv/SERV, Markit Wire, Acord, Oracle AIA-Insurance ...
- § Multiple usage scenarios:
 - Domain standards driven – often multiple standards
 - Partner driven – dominant partner sets standard
 - Application driven – dominant application sets standard
 - Bottom-up – legacy data content (including existing XML Schemas) dominates requirements
- § In-memory, “Big Data,” unstructured information, and dynamic schemas starting to play a role in some usage scenarios, at least on roadmaps

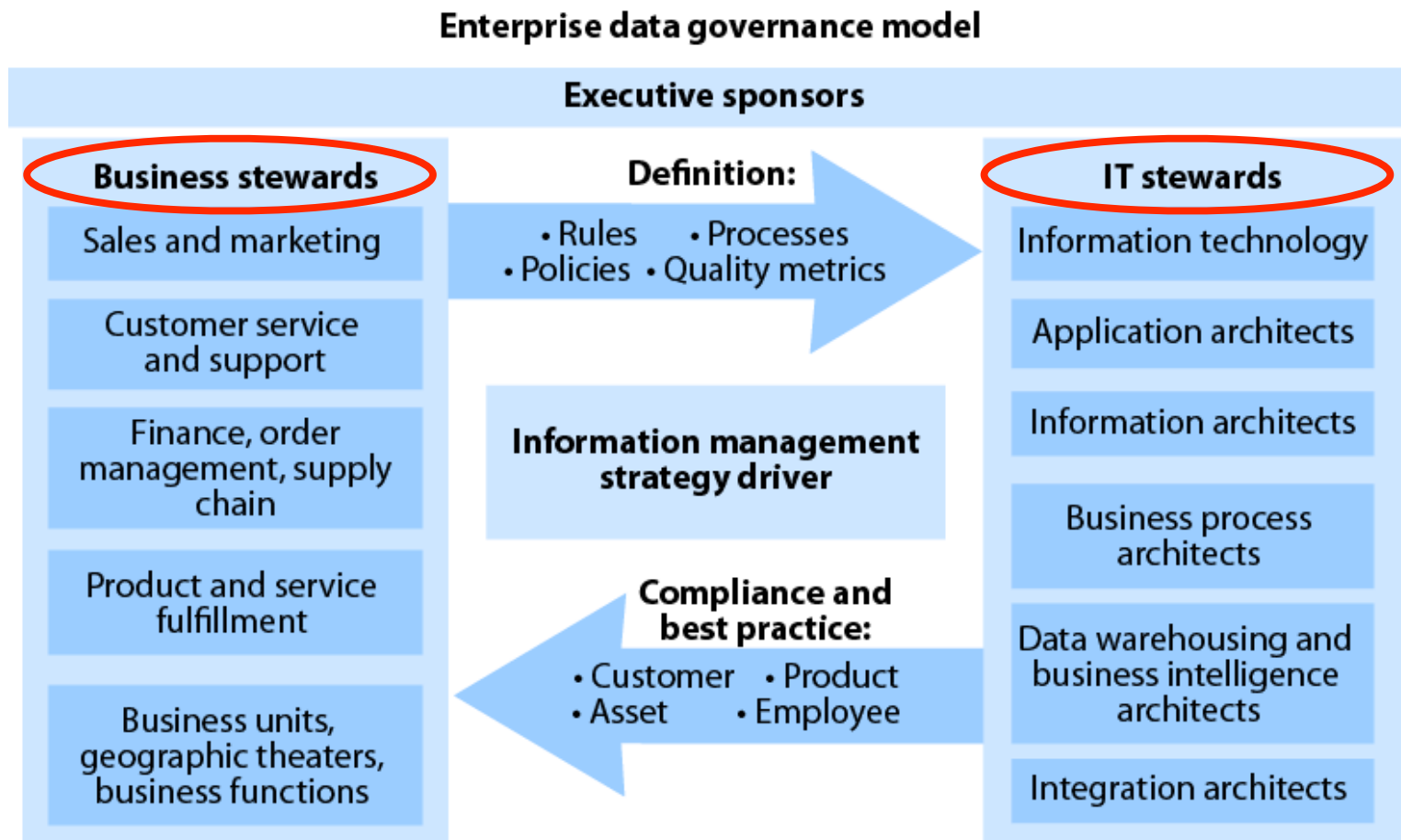
More lessons from current case studies

- § Tools are immensely helpful, especially when they automate model and data source discovery from myriad artifacts (especially in the bottom-up scenario)
- § Your model is worth *much* less if developers won't use it — so it must be **dead easy** for developers to consume & extend the model, and make use of it in their app dev environment
- § “Open,” *lightweight* tooling is best — allows substitution of tools, repositories, etc. — rather than a closed stack
- § Complex environments need multiple domain models, rich support for versioning and schema migration
- § Data services “access layers” are increasing in popularity and are driving increased need for *update* capability
- § Be careful about your implementation approach to minimize the impact of change on downstream apps

Minimize the impact of change, maximize flexibility

- § Reflect business domains in model domains (types of insurance business, personal vs commercial lines, agent-sold vs direct vs group, etc.).
- § Intersection set of all domains becomes agreed central common model.
- § Each domain derives off agreed central common model.
- § Each model (central or domain) has core plus extensions.
- § Core can reflect alignment with standards like ACORD – extensions model additional attributes specific to your business. Manage versioning impact from new standards versions with new model releases of core.
- § Domains have central core plus local (domain) core, with optional extensions from central and local models, as needed (usage specific).
- § Hold as little centralized power over the model as possible – it's expensive.
- § Empower domain owners as much as possible *consistent with your business model and culture* – it brings greater flexibility.

Who should “own” the canonical model(s)?



Source: August 11, 2009, “Refresh Your Information Management Strategy To Deliver Business Results”

Forrester report

© 2011 Forrester Research, Inc. Reproduction Prohibited

Example of centralized vs local empowerment

Large FS firm:

- § Centralized definition of core model, creation of adapters to integrate legacy (mainframe) apps to the ESB (and managing transformations *to* the canonical model)
- § Delegated to domains creation of adapters to integrate business-area apps to the ESB (and managing the transformations *from* the canonical model)
- § Centralized responsibility for managing change for core model (updates to standards bringing change impact to model, when chosen to update)
- § Delegated responsibility for managing change for model extensions, as needed to keep in synch with changing apps in the domains

Who should be the centralized group?

There are many potential candidates:

- § The Integration Competency Center
- § The Enterprise Architecture group
- § The Information Architecture group (inside EA, typically)
- § The team responsible for installing/maintaining the dominant app – group administration?
- § The DBA / Data Warehousing / BI group

Each has advantages/disadvantages depending on your history and local context

Important skills canonical modelers should have

In order of importance:

- § Business knowledge (*your* business, preferably)
- § Communication skills – a powerful advocate for the benefits of aligning with the canonical model, a salesperson, but flexible and results oriented
- § Modeling skill, good at abstract reasoning yet also good at explaining complex things to business people using models as *tools for understanding*
- § A “Get It Done” mentality – more interested in delivering working business solutions than in getting the perfect model
- § Empathy with developers – good to have worked as a developer and/or Chief Architect (responsible for delivery, not paperwork) in background

Canonical modeling trends — 2011 and beyond

- § Canonical models are becoming the model for access through shared data services “access layers” — with increased use of “in memory” data
- § The need for subsetting schemas and harmonizing back to the core schema is a significant factor in driving demand for CM tooling / automation
- § Combining structured/unstructured data is becoming a more mainstream, everyday pursuit — in some cases bringing “Big Data” approaches to bear
- § Integration with the Cloud is becoming more common, driving developers toward data services as the foremost way to access information — necessitating richer API support for developers (mainly REST based)
- § Federated repositories are becoming the rule rather than the exception – with some off-the-shelf integration
- § Data integration leaders are supporting common models and tooling across data services, data interchange, ESBs, B2B, BI/warehousing, and messaging

A related key trend for 2011 and beyond:



App platforms must become *elastic*.

Definition: Elastic application platform (EAP)

- An application platform that automates elasticity of application transactions, services, and data, delivering high availability and performance using elastic resources.

Elastic platforms are distributed

Availability

- Data can be replicated to 2 or more nodes
- Read-through and write-behind can save data to persistent data stores
- Cache can be configured to distribute locks to maintain data integrity during CRUD operations

Performance

- Data stored in-memory for fast access
- Data distributed across 2 or more nodes to avoid bottleneck
- Local cache can store data on same node as code for fastest access

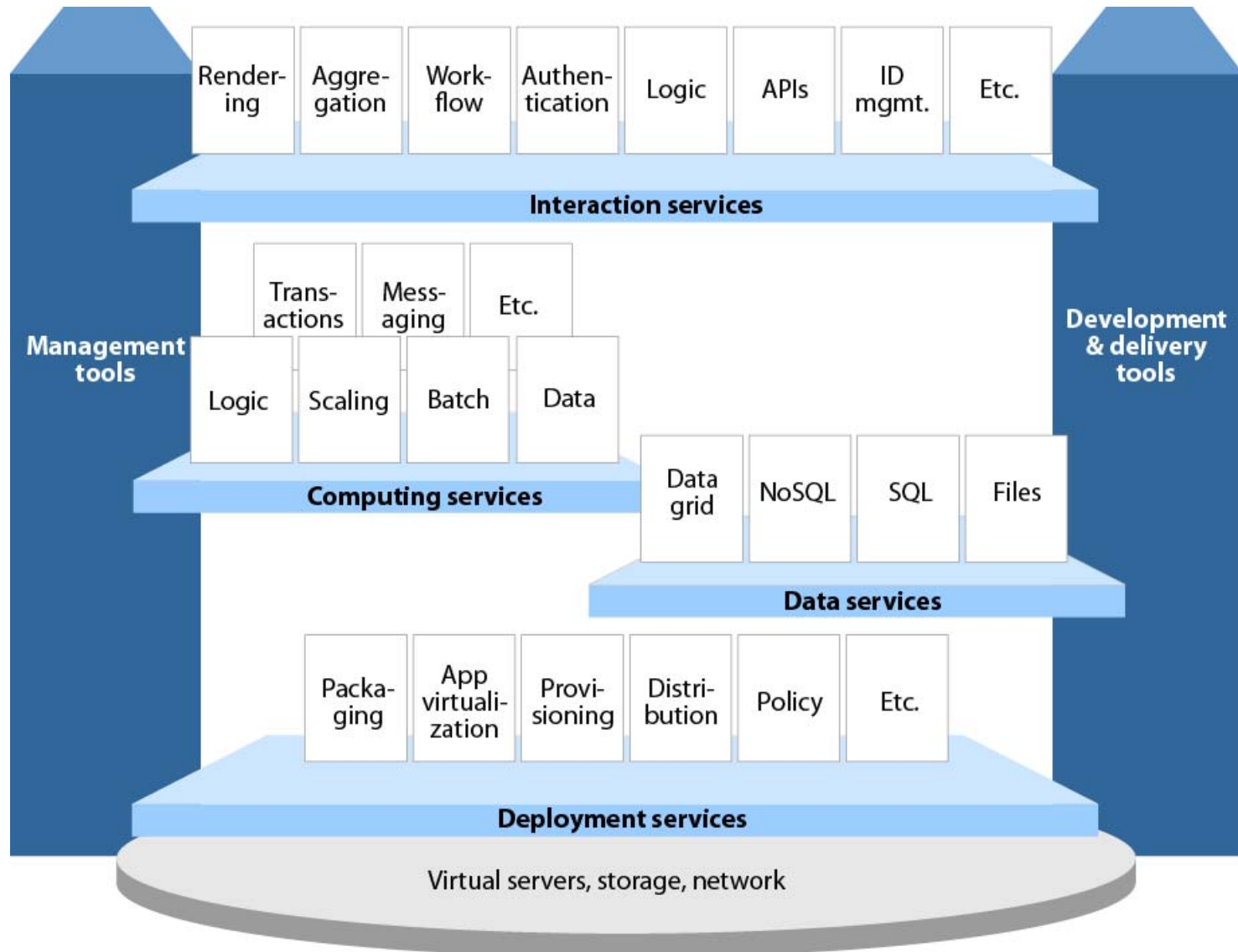
Scale

- Additional nodes can be added in run-time
- Data is re-balanced to take advantage of additional nodes



Elastic data can eliminate bottlenecks.

Elastic App Platform Reference Architecture



Recommendations:

- Link your canonical modeling strategy to broader information management goals.
- Use tooling, lifecycle integration, to drive productivity for modelers *and* developers.
- Design your information architecture for *agility*.
- Prepare for the coming “Big Data” revolution.
- Consider Elastic Application Platforms for even greater application agility, via Cloud.

Thank you

Mike Gilpin

+1 301.469.4863

mgilpin@forrester.com

Twitter: [@mgilpin](https://twitter.com/mgilpin)

www.forrester.com

FORRESTER