



CAiSE 2010, Hammamet, Tunisia
9 June 2010 14:30



Monitoring and Analyzing Service-Based Internet Systems through a Model-Aware Service Environment

Ta'iid Holmes¹, Uwe Zdun¹, Florian Daniel², Schahram Dustdar¹

¹ Distributed Systems Group, Institute of Information Systems
Vienna University of Technology, Vienna, Austria

² Information Engineering and Computer Science Department
University of Trento, Trento, Italy



Problem Statement

Service-Based Internet Systems become increasingly **complex**.

⇒ They become harder to

- manage,
- analyze, &
- monitor.

How to master the complexity?

How to monitor system requirements?

Context & Focus

Context: Model-Driven Development (MDD)

Focus: Compliance Monitoring of Business Processes

Models are used for describing the

- System

Process Models

- System Requirements

Compliance Concerns of Business Processes

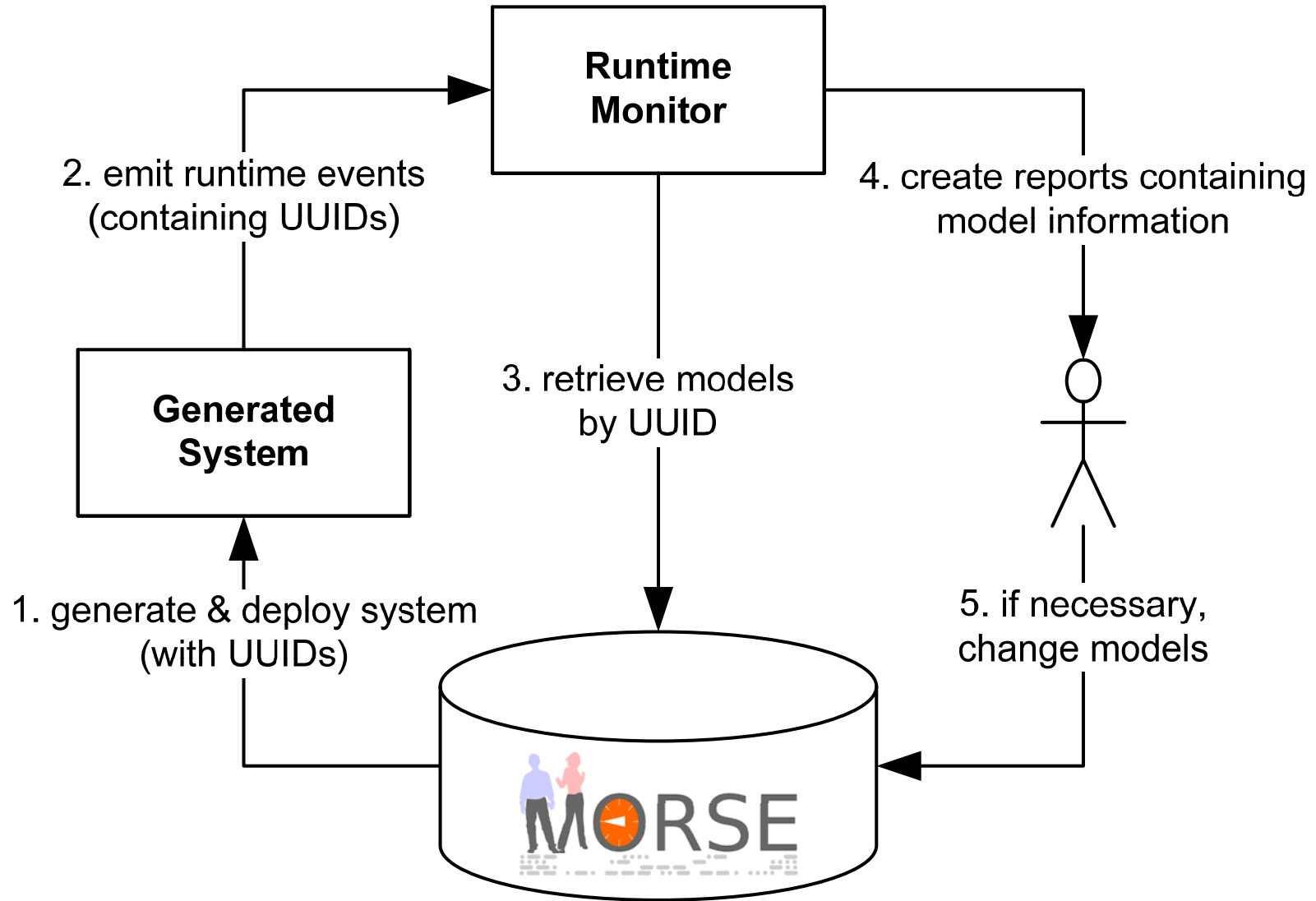
Requirement Models annotate System Models

⇒ *novel, direct linkage and correlation of
system & requirements models*

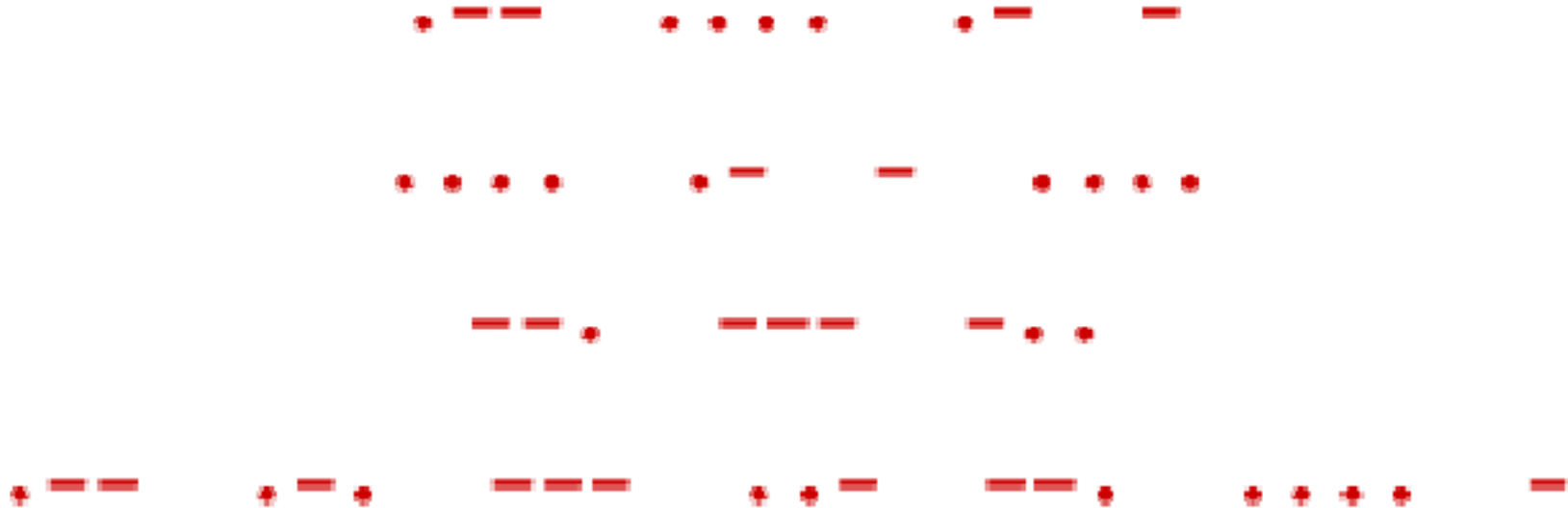
Runtime Challenges

- Traceability (high-level \leftrightarrow low-level model-instances and code)
 - essential for meaningful feedback from runtime to stakeholders and for identifying and understanding the root-cause
- Model-based execution is rarely used: missing infrastructure that supports dynamic lookup of models for model-aware systems

Approach | Overview



The First Electronic Message



Samuel F. B. Morse
May 24th, 1844



1. Model-Aware

because it stores models & MDD artifacts

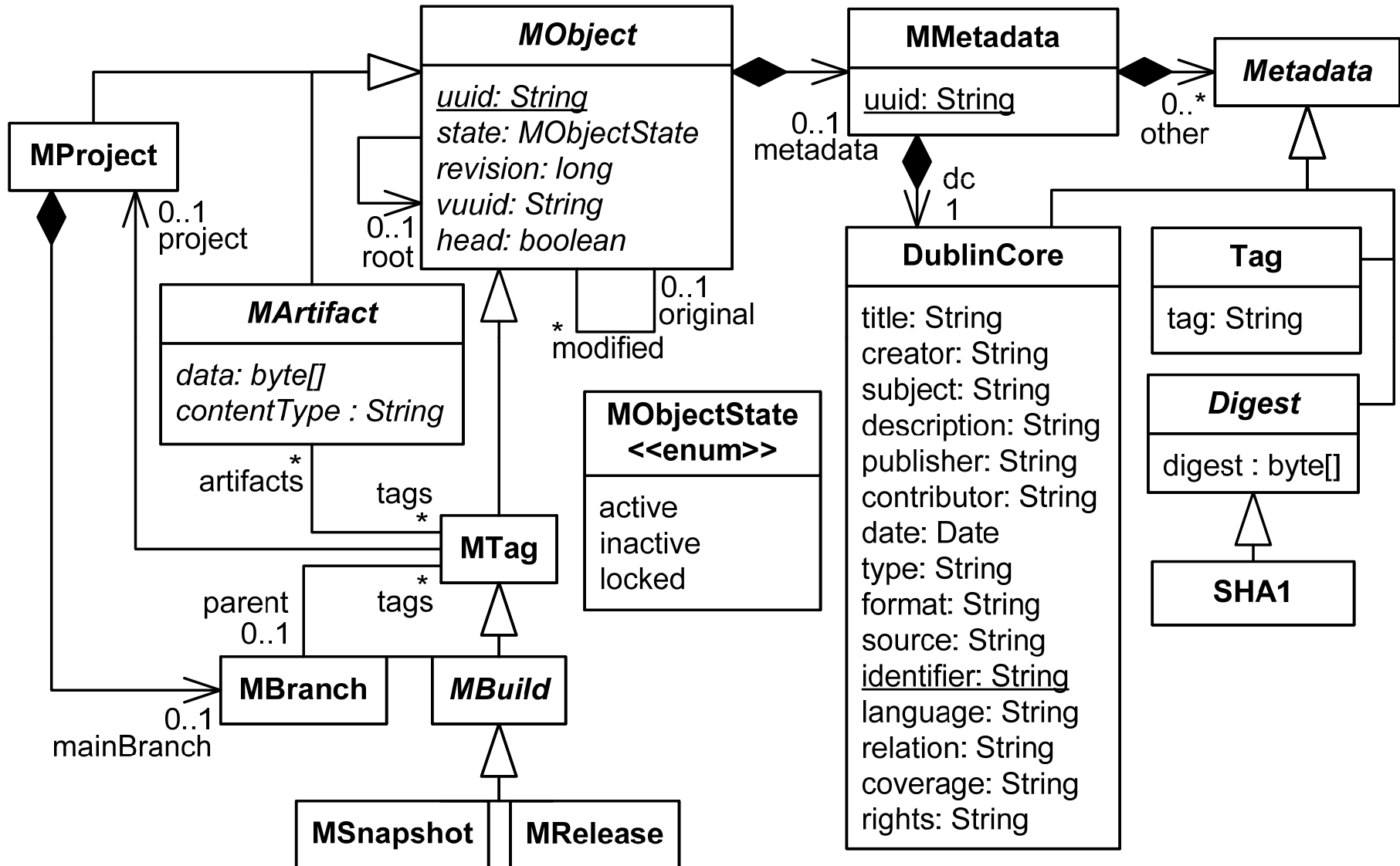
2. Repository

because it supports configuration management (e.g., versioning) of MDD projects

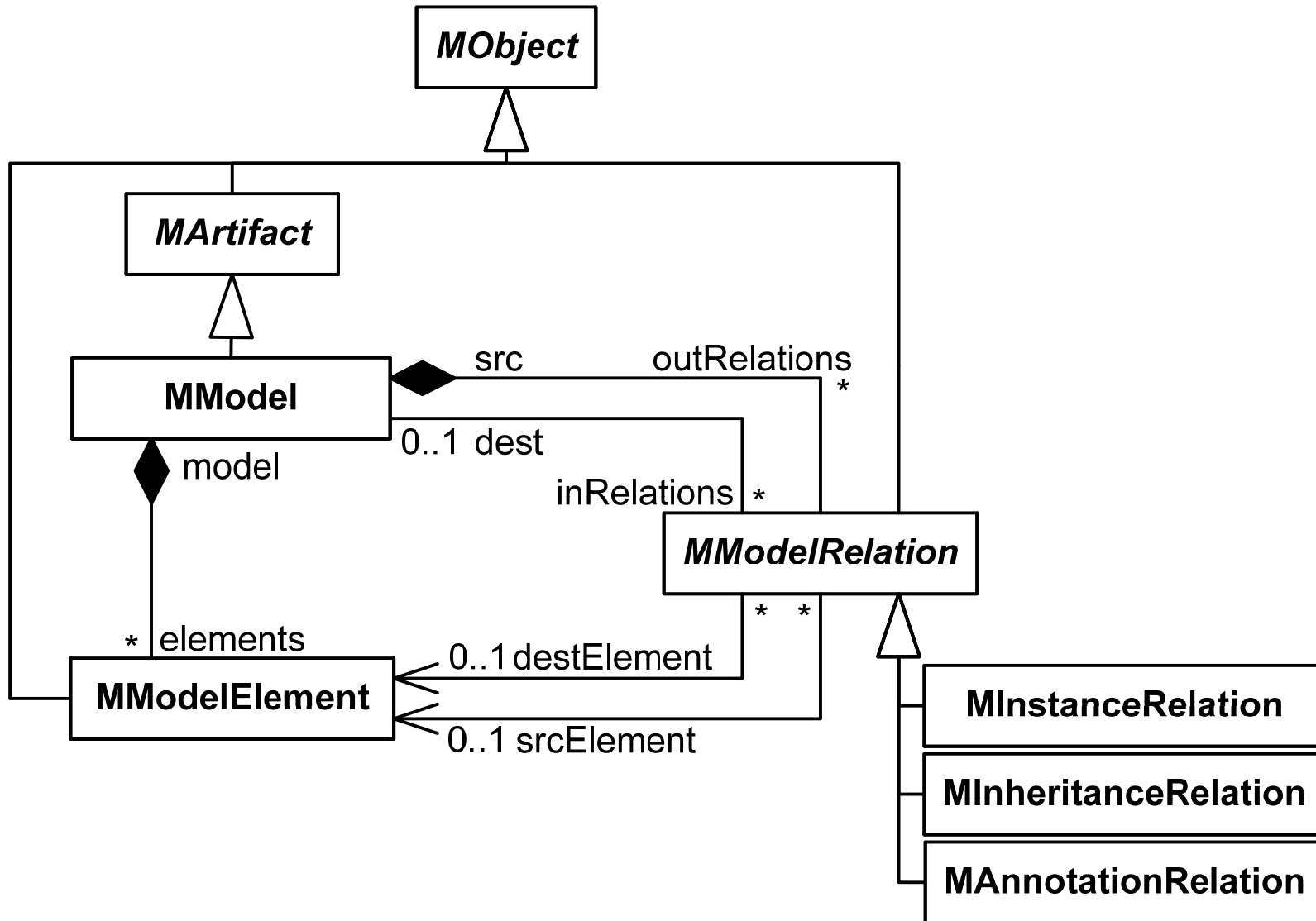
3. & Service Environment

because it offers service-based interfaces and integrates with other model-aware components, that cover the model-driven engineering lifecycle

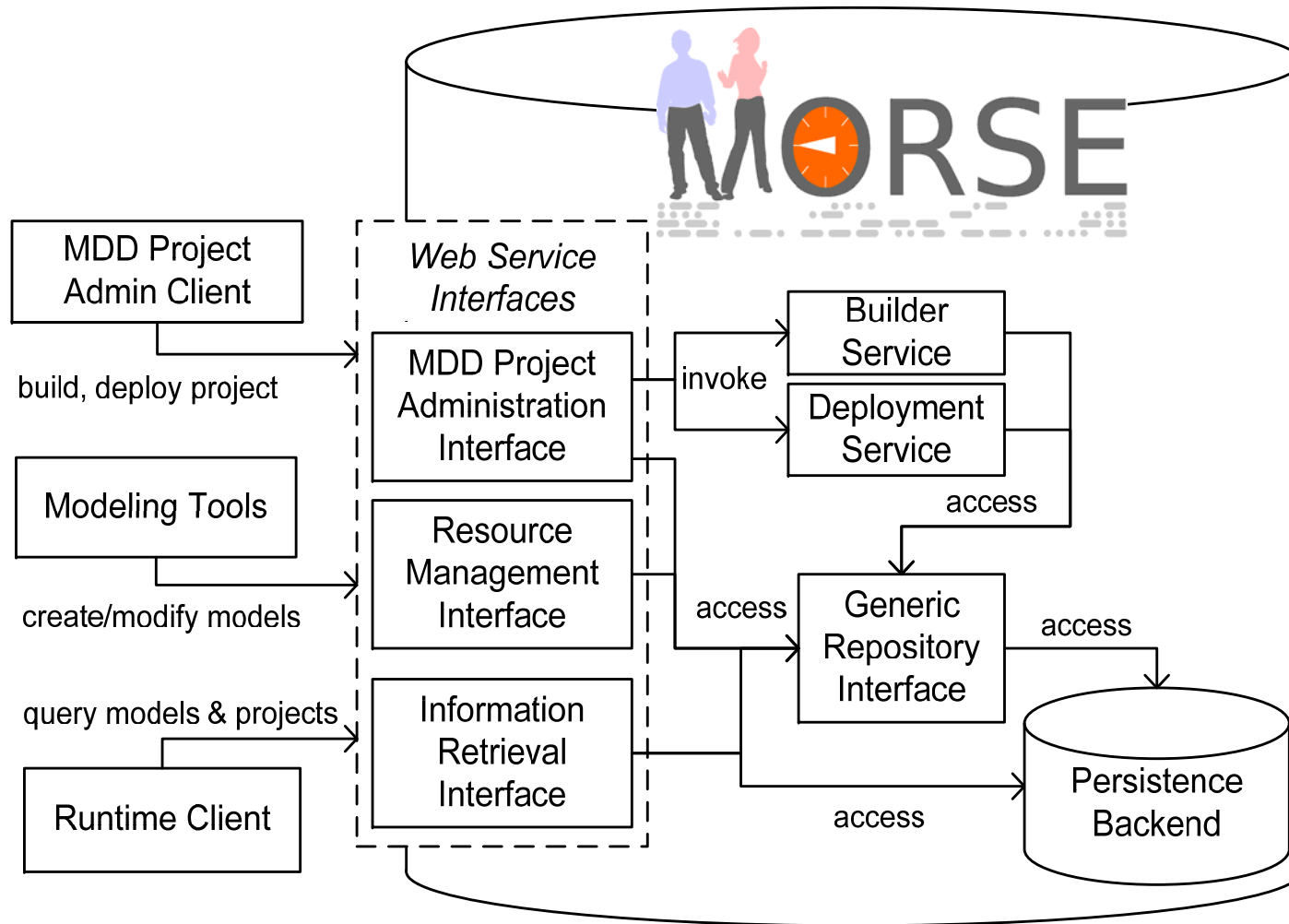
MORSE Core Model



MModel, -Element, -Relation



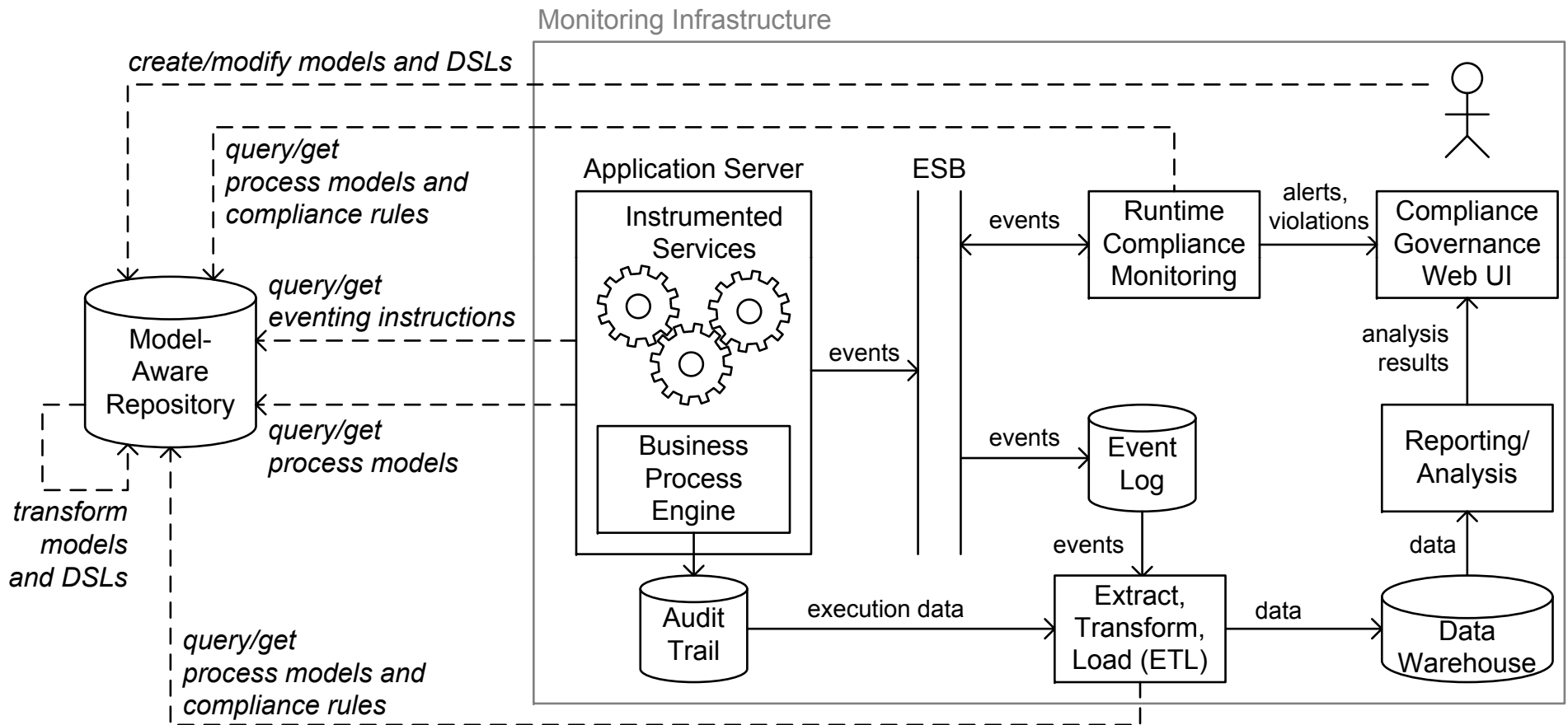
MORSE Architecture



Approach | Components

- Model-Aware Repository & Service Environment
 - services for the storage & retrieval of models
- Eventing Infrastructure
 - emits events with traceability information
- Monitoring Infrastructure
 - analyzes events to check compliance

Monitoring Architecture



MORSE Traceability

```
<morse:traceability build="56810150 - 5bd8-4e8e-9ec5-0b88a205946b">
  <row query="/process[1]"
    queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0">
    <uuid>6338b114-3790-4566-a5c4-a35aa4efe41b</uuid>
    <uuid>cd2865e2-73a7-4c8d-8235-974057a40228</uuid>
    <uuid>4bcf3d70-9c23-4713-8602-3b64160c45e8</uuid>
    <uuid>c568c290-e03e-46c8-9a9a-d7afde80cc3a</uuid>
  </row>
  <row query="/process[1]/sequence[1]/receive[1]">
    <uuid>354b5161-dfab-44ef-9d52-3fb6a9d3411d</uuid>
  </row>
  <row query="/process[1]/sequence[1]/invoke[3]">
    <uuid>7d32b4f4-4f63-4223-8860-db213f7e0fe1</uuid>
  </row>
</morse:traceability>
```

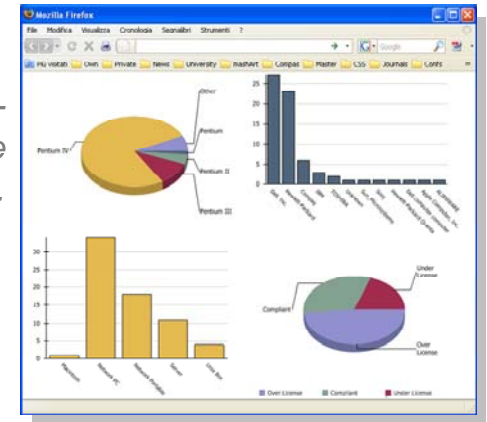
Case Study

SEC. 409. REAL TIME ISSUER DISCLOSURES.

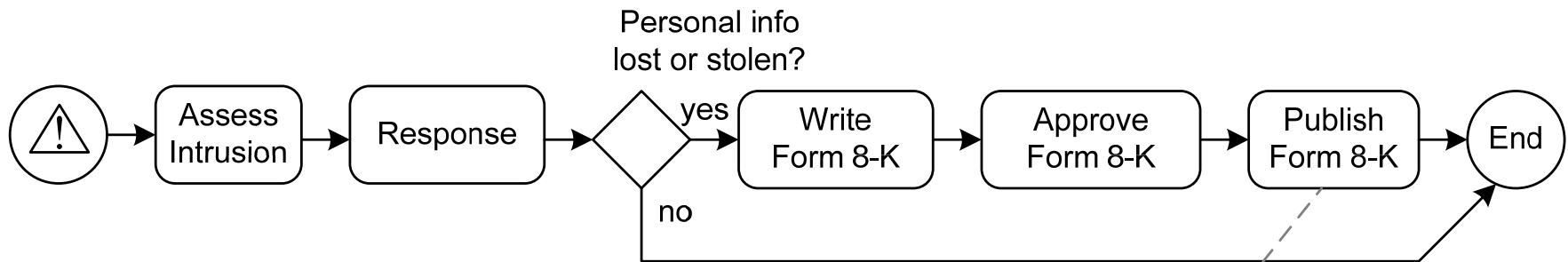
Section 13 of the Securities Exchange Act of 1934 (15 U.S.C. 78m), as amended by this Act, is amended by adding at the end the following:

“(1) REAL TIME ISSUER DISCLOSURES - Each issuer reporting under section 13(a) or 15(d) shall disclose to the public on a rapid and current basis such additional information concerning material changes in the financial condition or operations of the issuer, in plain English, which may include trend and qualitative information and graphic presentations, as the Commission determines, by rule, is necessary or useful for the protection of investors and in the public interest.”.

Dashboard with up-to-date compliance analysis reports.

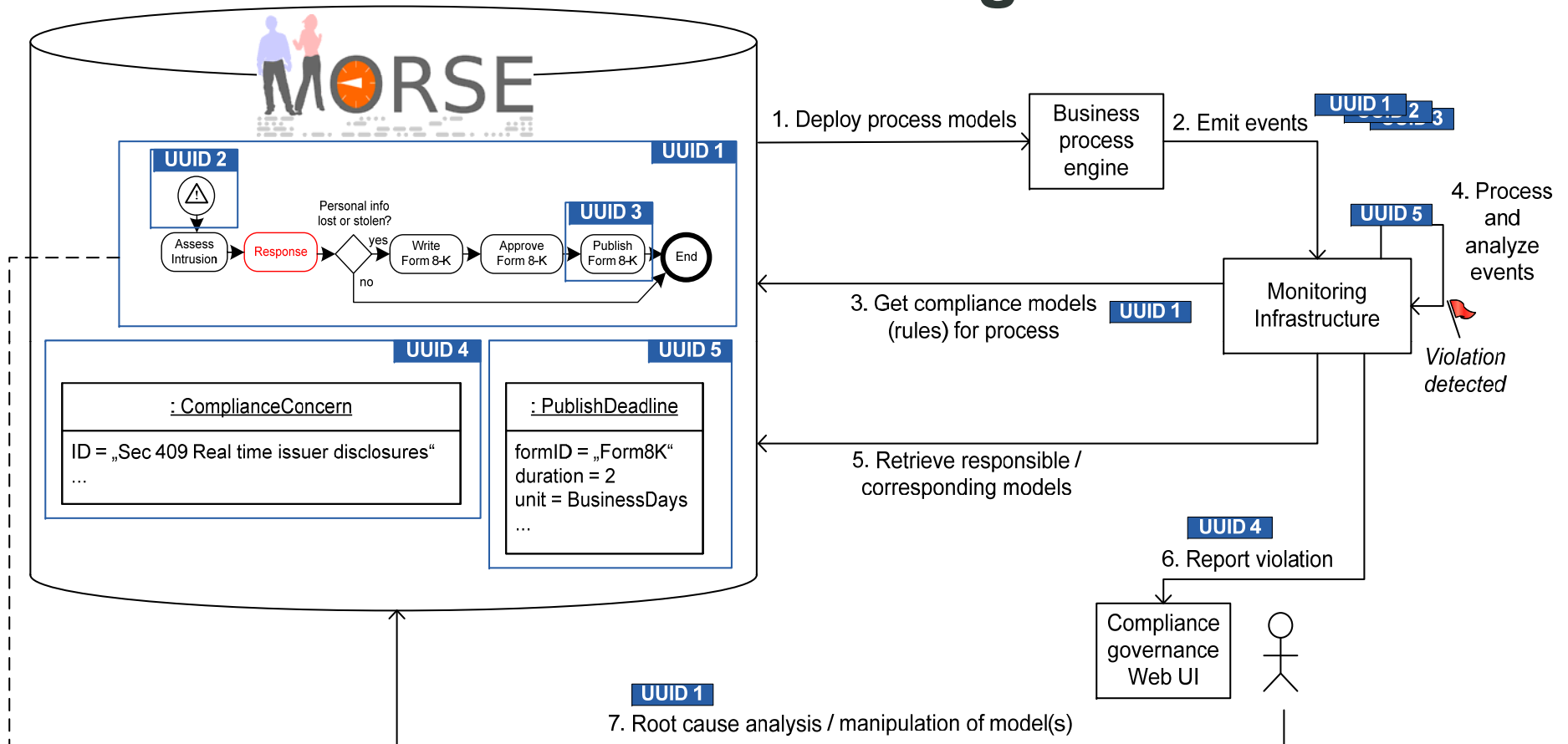


A credit card company might for instance implement a business process for the reporting of security breaches.



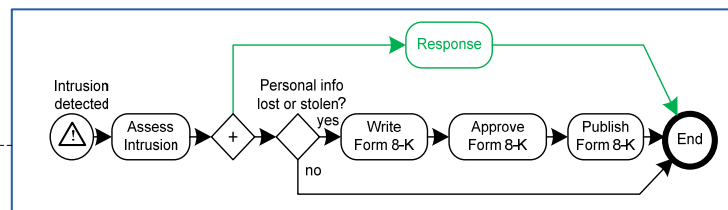
The publication of the press release must occur within 2 business days after the detection of the intrusion.

Root-Cause Analysis & Process Redesign

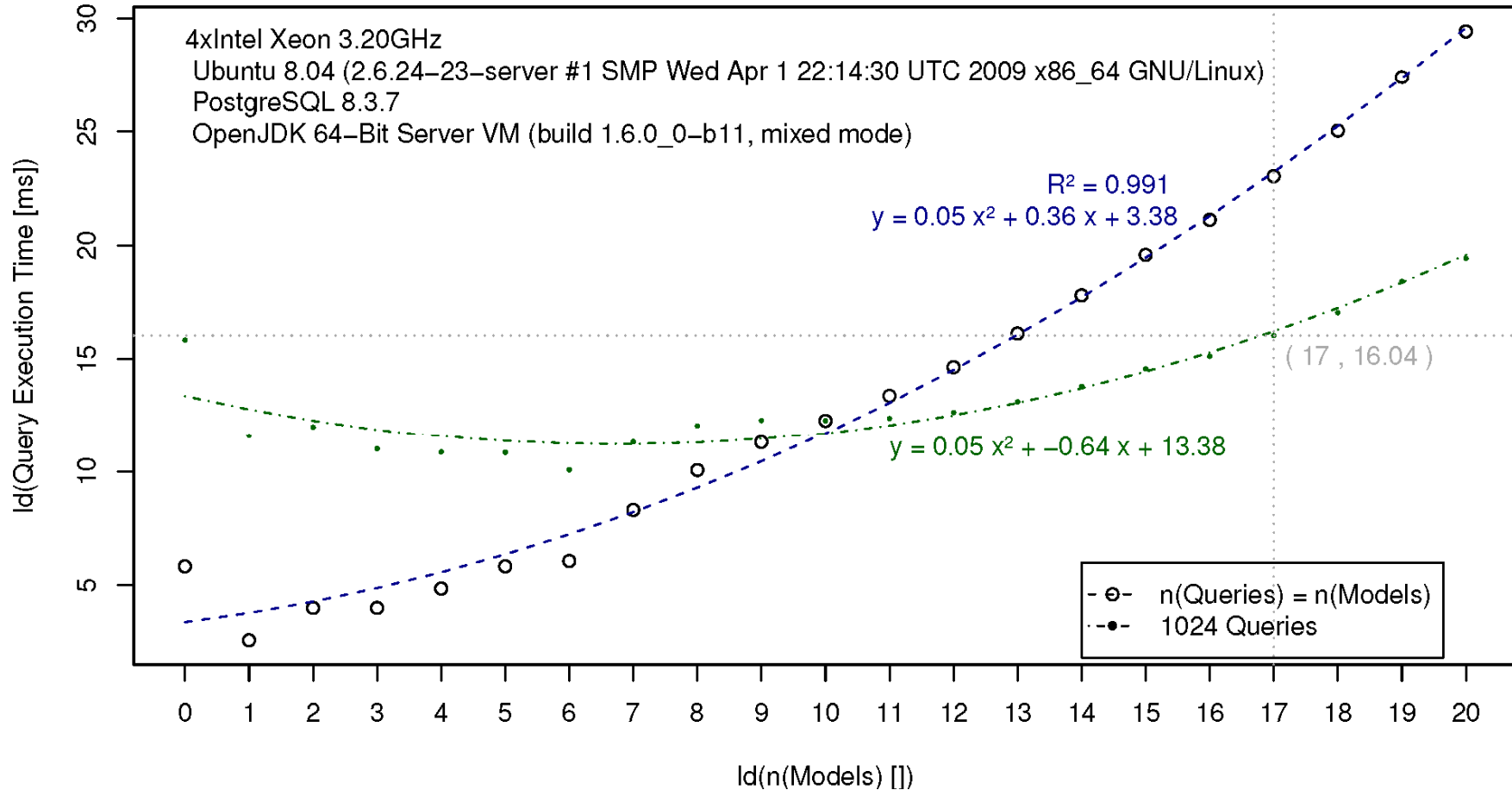


Before: sequential task execution; slow, lots of violations

After: parallel task execution; faster, fewer violations



Performance & Scalability



Conclusion

- correlated system & requirements models
- coupled MDD with eventing at runtime
- enabled dynamic, model-aware monitoring through MORSE
- shifted the model management from design time to runtime

Further Work

- adaptation, compensation, synchronization
- supporting evolution of models during runtime



Thanks for your attention!

Ta'id Holmes
Distributed Systems Group
Institute of Information Systems
TU Wien

<http://www.infosys.tuwien.ac.at>