

Community Systems Management Open Source “COSMOS” Creation Review

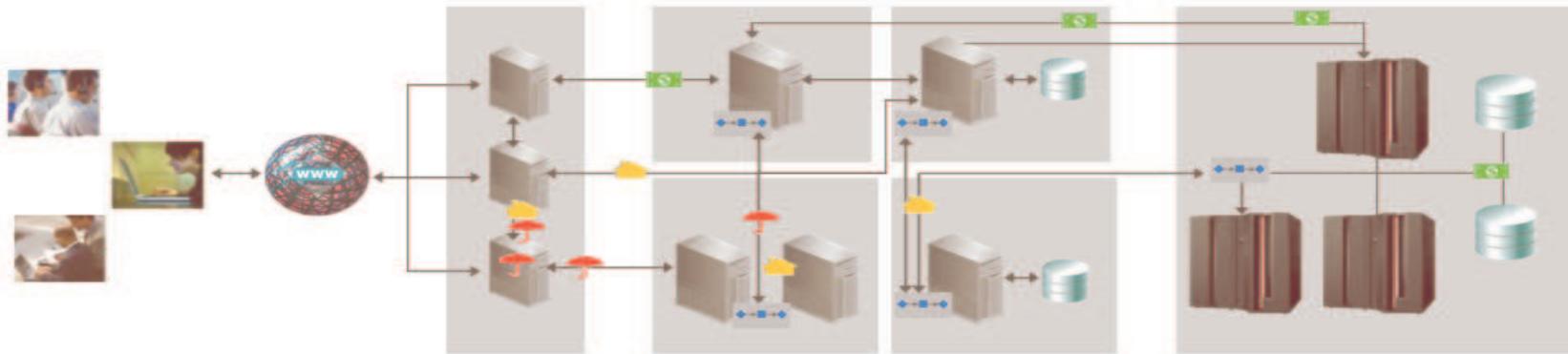
Put together by the COSMOS community
<http://wiki.eclipse.org/index.php/COSMOS>

Agenda:

- Motivation
- Overview
 - Key components
 - Initial implementation focus
- Structure and Operations

Challenges of IT Systems Management

Today's applications offer flexibility for business but introduce management challenges ...



“How do we coordinate problem resolution across all parts of the organization?”

“How can we minimize the disparate information we collect?”

“How can we instrument our systems?”

“How can we share resource descriptions in a consistent manner?”

“What industry standards can be applied to help solve the problems?”

“I know something’s wrong, but where?”

Project Overview

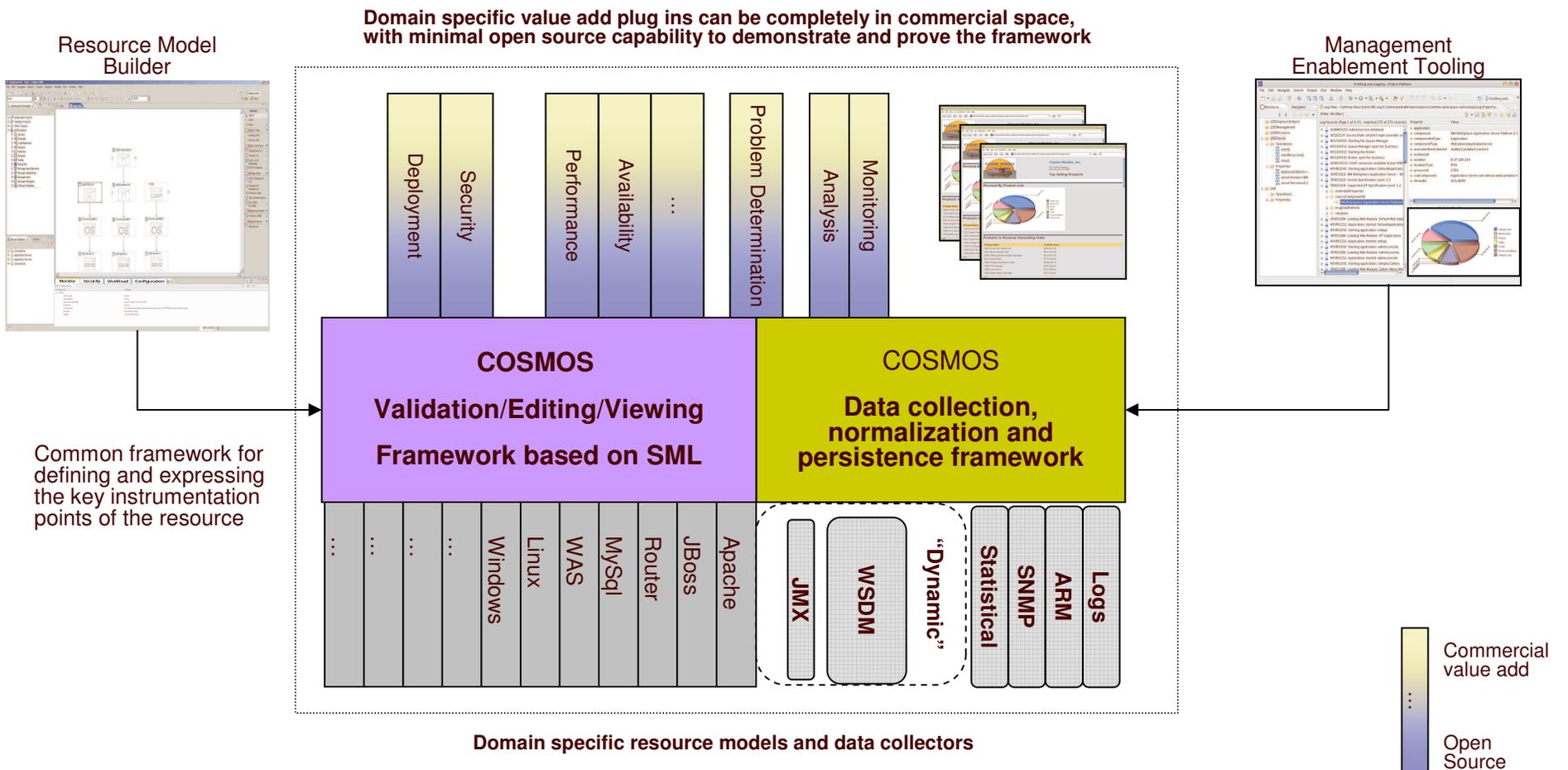
Community Systems Management Open Source*

- COSMOS project proposal submitted to Eclipse
 - Built upon the Eclipse Platform, the COSMOS project intends to develop generic, extensible, standards-based components for a tools platform upon which software developers can create specialized, differentiated, and inter-operable offerings of tools for system management
 - <http://www.eclipse.org/proposals/cosmos/>
 - <http://wiki.eclipse.org/index.php/COSMOS>
 - The framework will exploit standards-based services for monitoring, and modeling of enterprise resources, applications, and workloads
 - COSMOS will facilitate the next evolutionary move for systems management
 - COSMOS will define the common assets needed in a platform for vendors to derive immediate benefit and in turn provide higher value in their own offerings
 - COSMOS will facilitate a migration to a common and open standards driven set of components
 - COSMOS will define tooling for modeling resources based on SML standard (<http://www.serviceml.org/>)
 - In accordance with Eclipse process the COSMOS project will incubate following the guidance and supervision of the Eclipse Technology PMC
 - Expectation is COSMOS will graduate to a top level Eclipse project in 1H07

COSMOS Resource monitoring and modeling tools

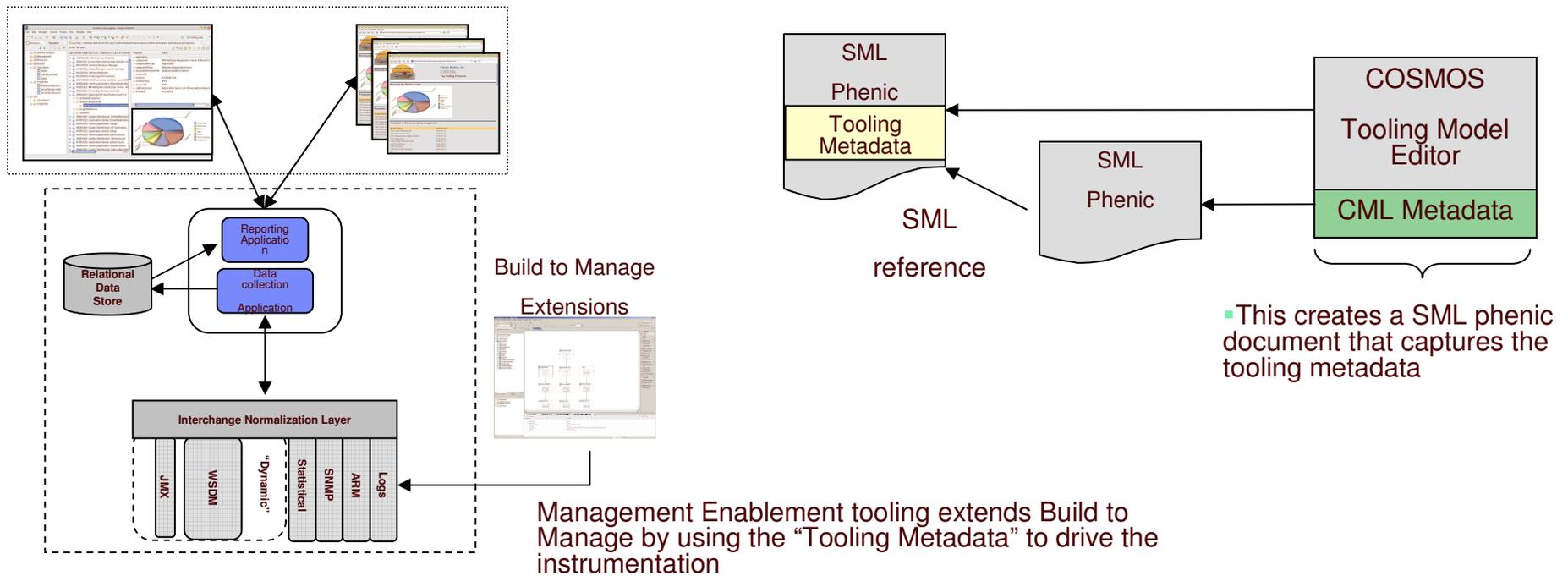
COSMOS is committed to providing exemplar usage of its APIs without eroding commercial opportunities.

The community will be end-to-end use case driven focusing on tooling to support production and the linkage to pre-production environment.



Future use of SML in the Management Enablement life cycle: Monitoring Domain, annotating what tooling is needed for a resource

- Management Enablement Tooling Builder
 - Defines how to construct tooling needed to observe the measurements described in the resource model
 - Based on resource model SML-IF documents
 - Extensible to use third parties' tooling template documents
 - SML template documents are used to build SML phenic instances; the result is an SML-IF document
 - Tooling metadata controls code generation and packaging



COSMOS: Key Components

- **Resource Modeling**
- Management Enablement
- Data collection
- Reporting

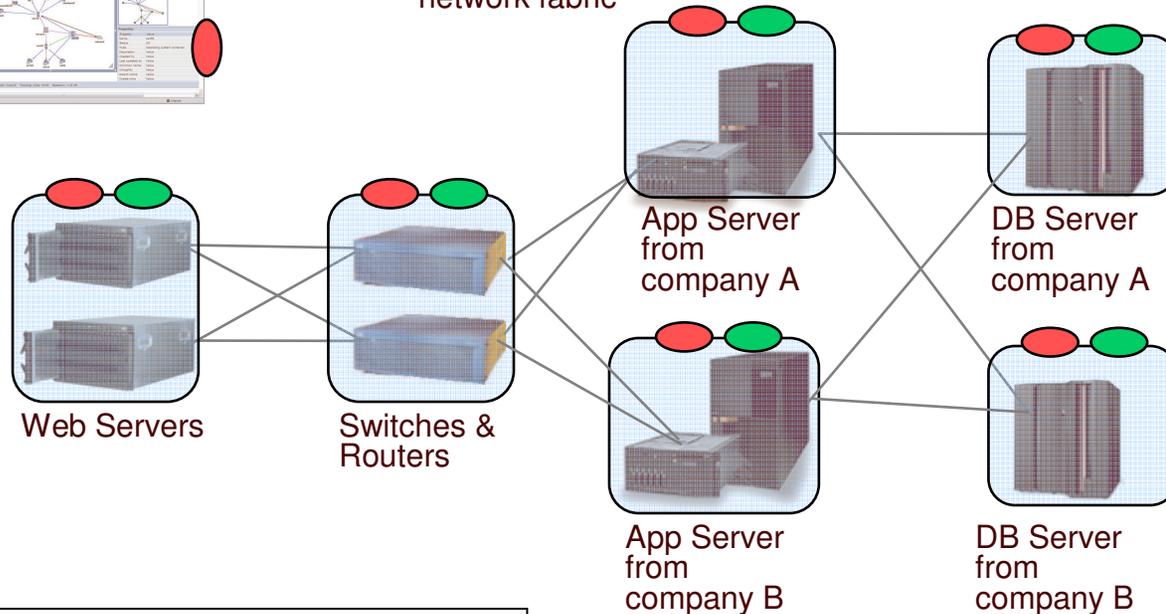
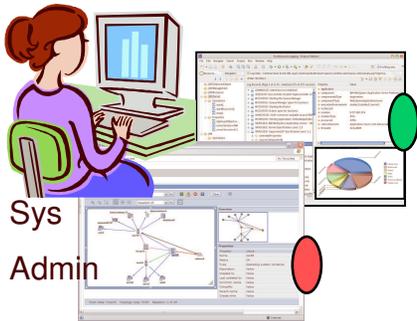
Current pain points

Each admin tool has a distinct way of representing the resources it can manage

Limited interoperability b/t tools (even those from the same company)

Limited co-existence of management tools at runtime

Management tools typically targeted at a specific resource domain e.g. network fabric



  Vendor Specific Management APIs

Customer Pain Points

- End-to-end problem determination is awkward and difficult
- Response to change in requirements is slow
- Creating higher value analytics difficult b/c of inconsistencies in the way resource information is expressed
- Inefficiencies in management costs displace development work

Often, this translates into custom instrumentation provided by management vendors.

This results in subtle, but important semantic differences in the way a resource is managed

Current pain points



Developer

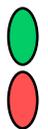
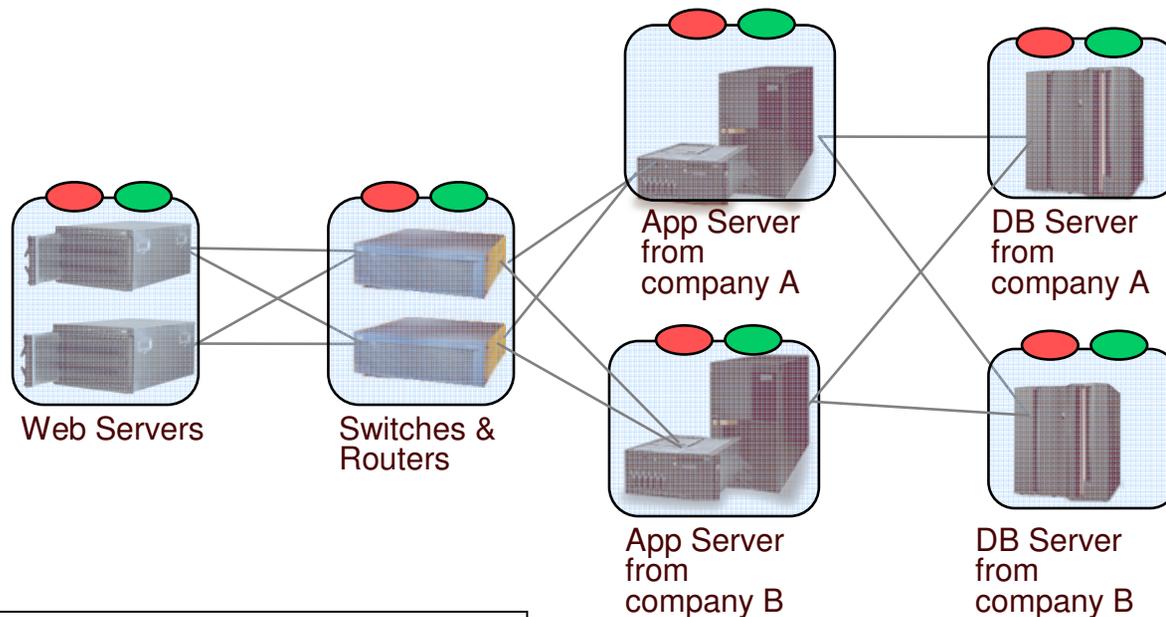
The developer has no consistent way of looking at all the resources in their environment

No consistent approach to describing the important management information

No standard way to share information with the sys admin

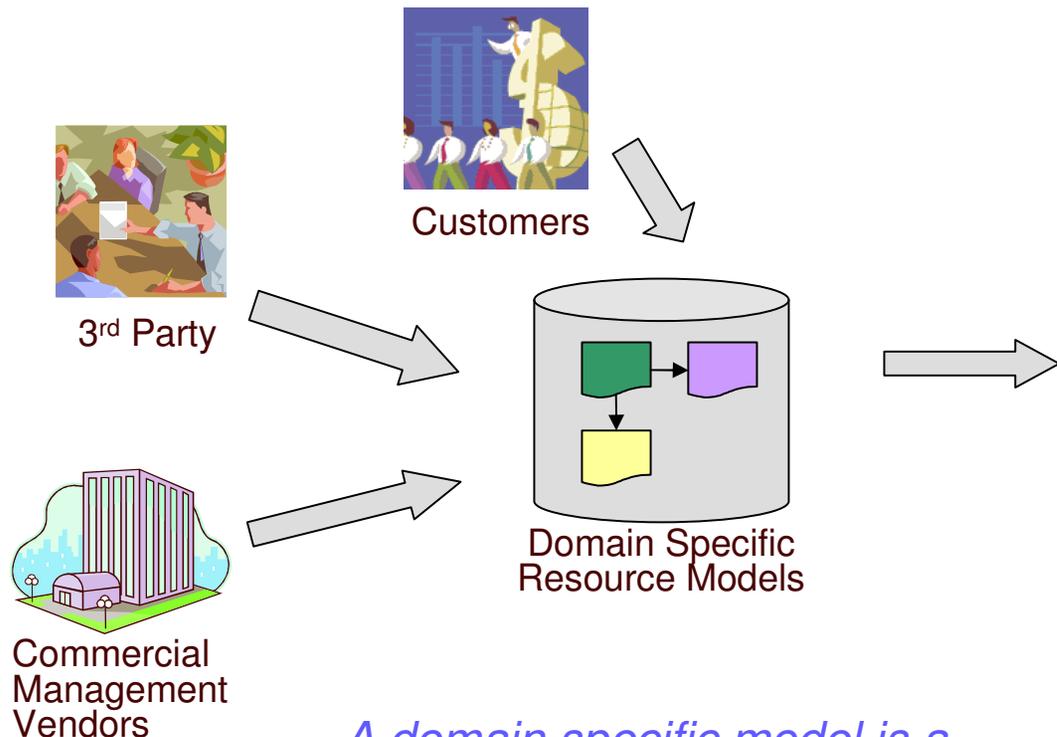
Customer Pain Points

- Difficult to extend existing instrumentation tooling for new roles or management disciplines
- New “kinds” of resources difficult to add



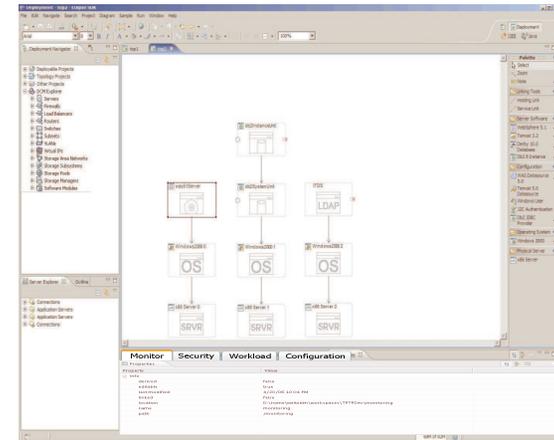
Vendor Specific Management APIs

Resource Modeling Landscape



A domain specific model is a fixed set of generic documents and phenic templates

Management Disciplines



- Deployment
- Monitoring
- Security
- Problem Determination
- Performance
- Availability

Each discipline specific tool deals with additional generic constraints and phenic instances

Resource Model Value

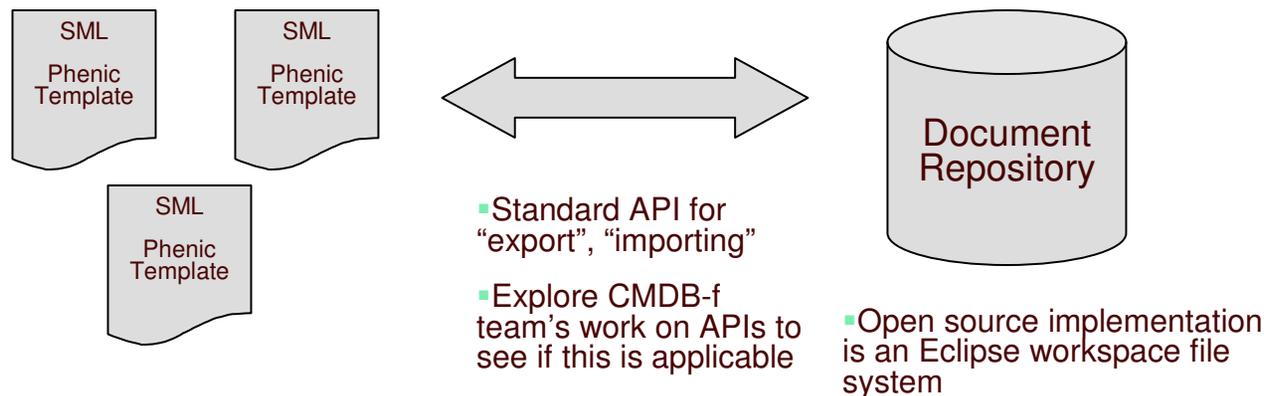
- **Value to Resource Model providers**
 - Decouples the management tools from the resource providers
- **Value to domain model consumers**
 - Decouples from resource provider
- **Value to end user**
 - Vendor neutral, extensible, tooling based upon resource models
 - Deployment and Configuration of resources
 - Simple tools to create complex models
 - Integration of Root Cause & Problem Determination

Requirements for Resource Model Ecosystem

- **Value increases exponentially with # resource models**
 - Enable creation of domain specific models
 - Demonstrate generic consumption as a building block
- **To make the models interoperable, they must be extensible**
 - Must be easy to extend
 - Must provide exemplary framework to enable rapid creation of role specific tooling

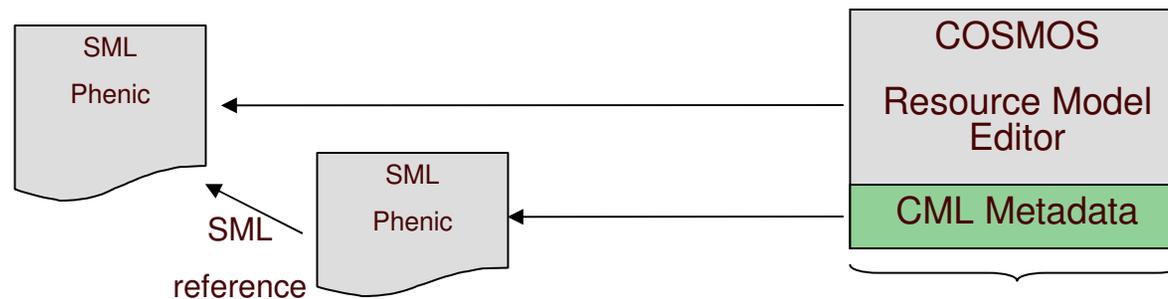
COSMOS deliverables related to the creation of domain specific models

- Tooling for validating SML instances and SML-IF documents
- Tooling for importing/exporting SML-IF documents to and from predefined repositories.
 - Import and export capabilities should be extensible so that consuming products can produce implementations to deal with custom repositories
 - Open source Document Repository is implemented as a file system structure
- Tooling for creating SML template documents
 - an SML template document is an SML instance defining a common pattern that can be re-used and adapted in different domain models
- Tooling for creating domain models based on existing SML templates (Resource model builder)
 - The tool should be extendable to allow registration of third parties SML templates
- Extensible architecture to allow registration of new domain models or extensions to existing models



Using SML in the Monitoring life cycle: Annotating what can be observed on a resource

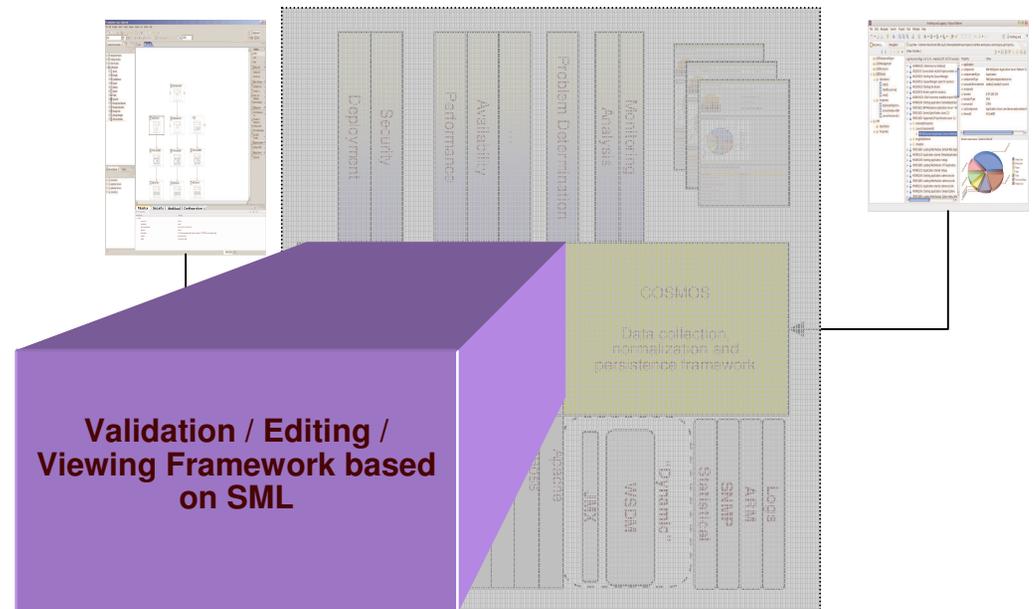
- Resource Model Builder
 - Based on a set of SML template document (CML metadata)
 - Extendable to use third parties template documents
 - SML template documents are used to build SML phenic instances; the result is an SML-IF document



- This creates a SML phenic document that captures the monitoring metadata
- This is a CML proposal**

Release 1 Deliverables SML

- Import of SML-IF into workspace
- Export of SML-IF from workspace
- Validation in context of workspace
- Edit document in context of workspace



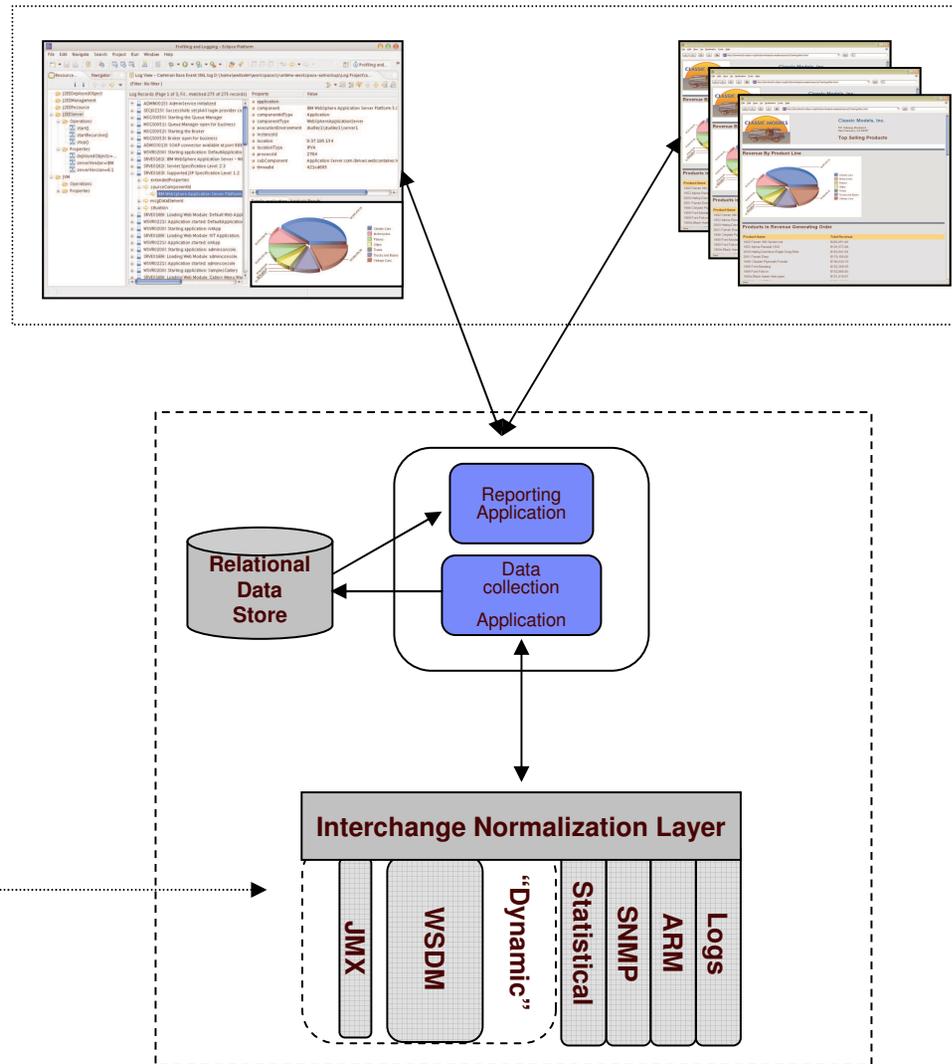
Current information available at:

http://wiki.eclipse.org/index.php/CosmosResourceModelingComponent#Resource_Modeling

COSMOS: Key Components

- Resource Modeling
- Management Enablement
- Data collection
- Reporting

Using Management Enablement in the Monitoring life cycle:



Commercial monitoring systems can:

- Reuse COSMOS data collectors

Release 1 Deliverables Management Enablement

- **Library of probes driven by use case**
 - By default integrated with TPTP
 - Software under consideration
 - Tomcat, MySql, Derby, JBoss, SNMP
- **Integration with data collection control**

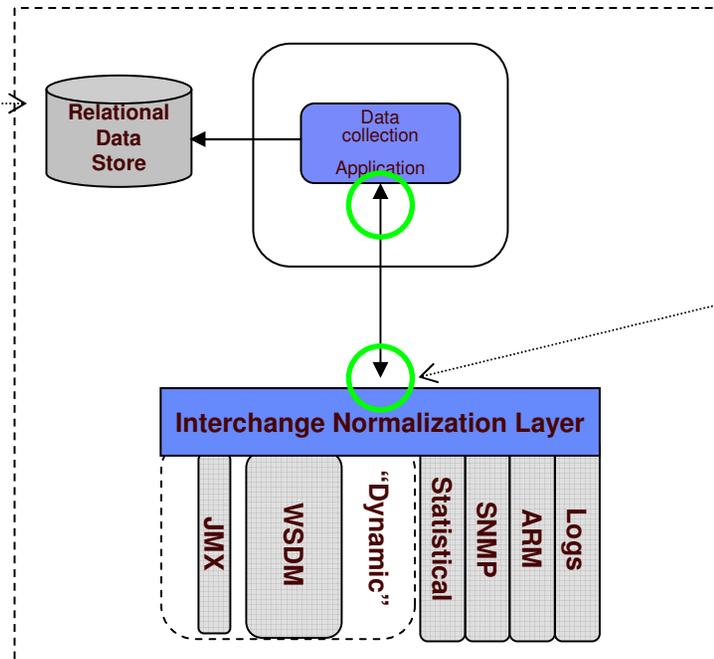
COSMOS: Key Components

- Resource Modeling
- Management Enablement
- **Data collection**
- Reporting

Using Data Collection in the Monitoring life cycle:

Commercial monitoring systems can:

- Extend their data by leveraging the persistence service



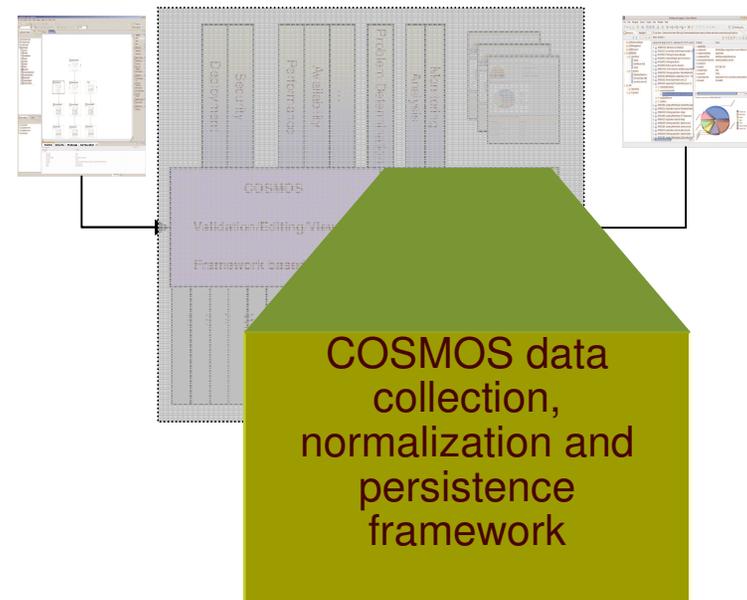
- API for dynamic data collectors that support two way communication.

 Declared API

Release 1 Deliverables

Data Collection

- **The definition of a persistence service with (shared w/TPTP)**
 - Persistence API
 - Query API
 - Data Collection Control API
- **Adapters for data collectors into persistence API**
- **Database schema**



COSMOS: Key Components

- Resource Modeling
- Build to Manage (BtM) a.k.a Enabling Management
- Data collection
- **Reporting**

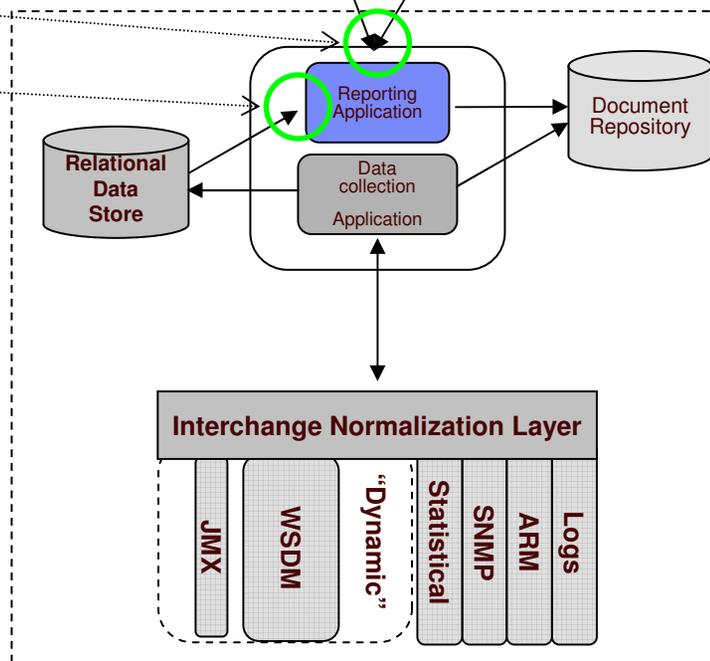
Future use of SML in the Monitoring life cycle: Reporting the observation

Commercial monitoring systems can:

- Reuse community created reports
- Extend their data by leveraging reporting API



- Use BIRT ODA to connect to Reporting API
- Define reporting API



Use phenic documents to indicate

- What to collect
- What information is observable
- Phenic documents used to carry the data

Commercial monitoring systems can:

- extend their data by leveraging the document repository

COSMOS Resource monitoring and modeling tools Reporting – Release 1 Deliverables

- **Web Service APIs for data access**
- **BIRT ODAs for access to relational data through web service APIs**
- **BIRT Reports for**
 - Resource status over time
 - Performance metrics over time
 - Current status overview



COSMOS

Operational Structure

Cross Project Linkage

- **A more detailed look at the linkage is on the COSMOS wiki:**
 - <http://wiki.eclipse.org/index.php/COSMOSProjectRelationships>
- **TPTP**
 - Dependency on data collection and access api work
 - Includes data collector adaptors
 - Dependency on build to manage tool output as an enabler
- **BIRT**
 - Dependency on BIRT for the reporting project
- **WTP**
 - Future investigation regarding deployment modeling
- **STP**
 - No direct dependencies near term
- **EMF**
 - No direct dependencies near term. Going forward the Type Builder will be based on GMF and as a result, on EMF, GEF
- **GMF**
 - Resource Modelling tooling such as the Model Editor and Template Editor, will be eventually based on GMF, GEF, EMF

Leadership Team

- **Management Committee**
 - Each top level component lead
 - AG, PG, RG Leads
 - Marketing Group lead
 - Toni Drapkin (acting lead) VOTED in by the team on 11/22
 - TBD (co-lead)
- **Architecture**
 - All committers
 - Mark Weitzel (proposed lead)
- **Planning**
 - Volunteer representatives from contributing community (maximum one per company)
 - TBD (proposed lead)
- **Requirements**
 - Volunteer representatives from contributing community (maximum one per company)
 - TBD (proposed lead)
- **Marketing Group**
 - Volunteer representatives from contributing community (maximum one per company)
 - Mark Morneaut, IBM (proposed lead)
 - Judy Schramm, OC Systems

COSMOS Project Release 1 Milestones: 2006-2007

2006

- **September:**
 - Kickoff in Eclipse and gain agreement on project charter
- **October:**
 - Bring community together
- **November:**
 - Begin work on
 - SML based tooling
 - Data collection
 - Data reporting
 - Build to Manage
- **December:**
 - Begin incubation under guidance of Technology PMC
 - Establish infrastructure within Eclipse

2007

- **January:**
 - Review SML tooling prototype at SML interoperability conference with partners
- **March:**
 - COSMOS Release .5 Availability
 - SML based tooling
 - Data collection & visualization
- **June:**
 - COSMOS release 1.0 Availability
 - Monitor UI
 - Remainder of Data Collection
 - Additional SML tooling updates
 - Release alongside Europa
- **1H07:**
 - Graduate/Launch COSMOS as Top level project

Comments about the community response to the proposal

The community is ready to move forward and is supporting project creation!!

- The community has demonstrated support for COSMOS by actively participating in its definition
 - Community wiki (hosted at Eclipse) containing
 - High level architecture
 - Use Cases
 - Design Discussions
 - Eclipse Con Submissions
 - Active newsgroup (over 70 postings)
 - Submissions to Eclipse Con 2007 on Cosmos
- 7 Companies participated in weekly calls to define the scope and deliverables
 - 4 prepared to provide committers

Initial Community

The following companies have actively participated in shaping the project's scope and initial deliverables.

- Alterpoint (www.alterpoint.com)
- Cisco (www.cisco.com)
- Compuware (www.compuware.com)
- GroundWork (www.groundworkopensource.com)
- IBM (www.ibm.com)
- OC Systems (www.ocsystems.com)
- Sybase (www.sybase.com)

Initial List of Committers

Name	Organization	E-Mail	Read Development Process*	Component
				Web Content Access for everyone
Cole, Oliver	OCSystems	oc@ocsystems.com	Y	Management Enablement
Drapkin, Toni	IBM	tdrapkin@us.ibm.com	Y	Web Content
Ebright, Don	Compuware	Don.Ebright@compuware.com	Y	Data Collection
Jerman, Steve	Cisco	stjerman@cisco.com	Y	Resource Modeling
Lee-Loy, Sheldon	IBM	sleeloy@ca.ibm.com	Existing committer	Reporting UI
Mehregani, Ali	IBM	amehrega@ca.ibm.com	Existing committer	Resource Modeling
Popescu, Valentina*	IBM	popescu@ca.ibm.com	Existing committer	Resource Modeling
Schramm, Judy	OCSystems	jas@ocsystems.com	Y	Web Content
Sluiman, Harm*	IBM	sluiman@ca.ibm.com	Existing committer	Resource Modeling
Thomas, Craig	Groundwork	cthomas@groundworkopenseurce.com	Y	Reporting
Weitzel, Mark	IBM	weitzelm@us.ibm.com	Existing committer	Management Enablement

****Project Mentors: Harm Sluiman & Valentina Popescu (Voted in by team 11/12)***

*Confirmation that the project members have read and understand the Eclipse Development process and these guidelines:
http://wiki.eclipse.org/index.php/Development_Process_2006_Revision

References

- **Eclipse Development Process and Guidelines**
 - <http://www.eclipse.org/legal/newcommitter.php>
 - http://www.eclipse.org/membership/become_a_member/committer.php
 - http://wiki.eclipse.org/index.php/Development_Conventions_and_Guidelines

Glossary

- **SML validation**
 - The action of validating SML extensions which are provided as additions to the xml 1.0
 - Schematron validation
- **SML-IF instance validation**
 - The process of validation the content of an SML-IF resource
 - Validate SML phenic documents contained by the SML-IF resource
 - Apply any schematron rule defined as a genic document
- **Template document**
 - An SML instance defining a common pattern that can be re-used and adapted in different domain models
- **Domain models**
 - The root of an SML-IF document. Contains a set of phenic and genic documents
- **Resource domain**
 - A set of genic and/or template documents that can be used to build define a domain
- **Template editor**
 - An editor that can create genic and template documents
- **Model editor**
 - An editor that can create domain model instances, based on a set of predefined templates
- **Genic Documents**
 - The subset of documents in a model that describes the schemas and rules that govern the structure and content of the model's documents. This specification defines two kinds of genic documents - XML Schema documents that conform to SML's profile of XML Schema and rule documents that conform to SML's profile of Schematron.
- **Phenic Documents**
 - The subset of documents in a model that describe the structure and content of the modeled entities.

Note: The most current copy of the glossary is on the wiki:
<http://wiki.eclipse.org/index.php/Glossary>