



App Work Group Current work

October 2002

QOS App Manageability discussion

- This is a collection of slides to aid the discussion of work within app group

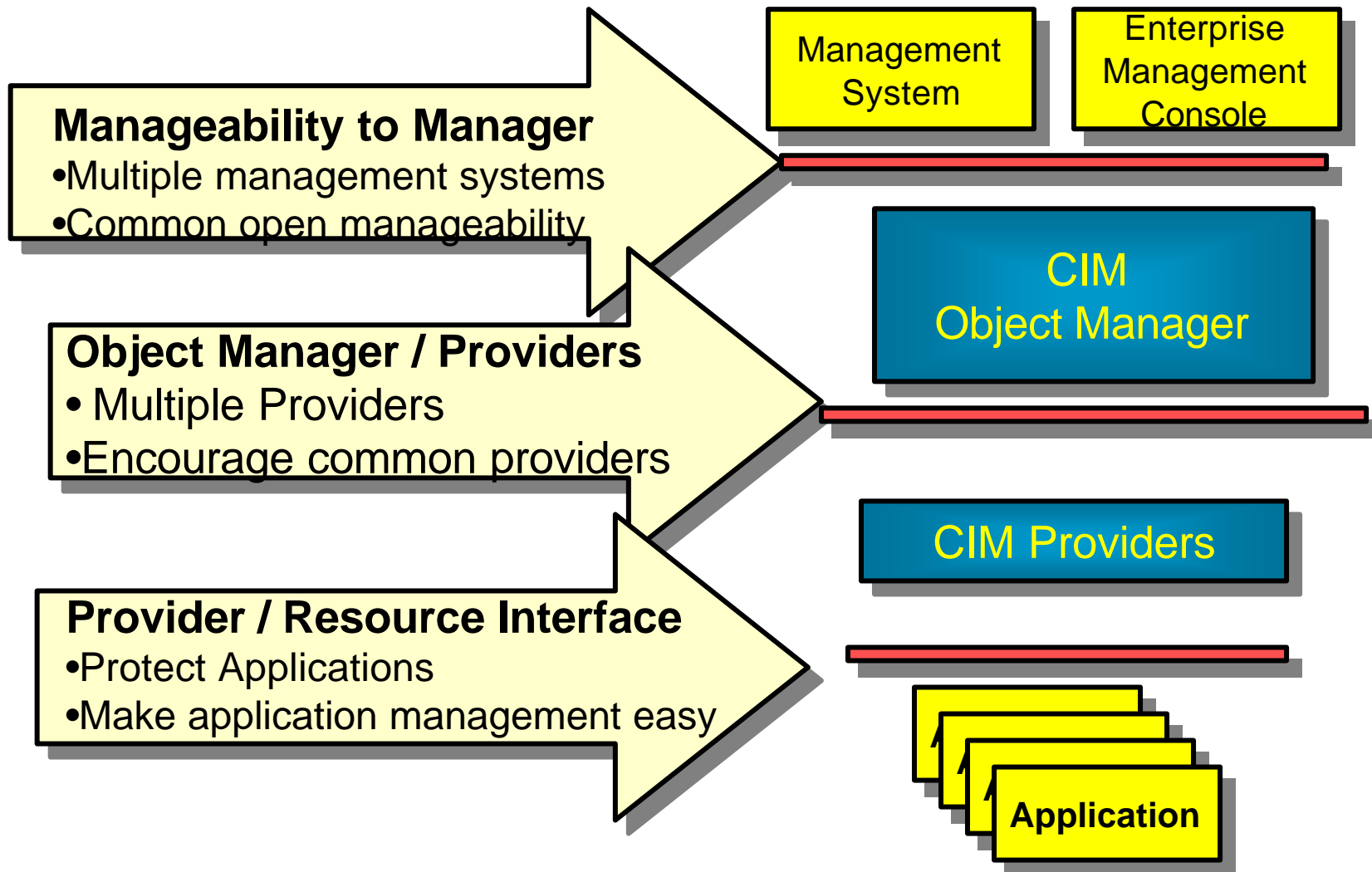
Background

THE *Open* GROUP

- ❑ App work group set up as joint DMTF / TOG working group.
- ❑ Aimed now at general issues of app management
- ❑ Creating models for information to do application management.
- ❑ Four directions
 - Bottom up – Creating the tools (metrics, state concept, operations to support app management within CIM model
 - Instrumentation and mapping the instrumentation to models
 - What are the real objectives of management in this application
 - The models themselves to represent the different major views of running applications that may be wanted by management systems.

Key Interoperability Interfaces

THE *Open* GROUP



Management Standards Today

THE *Open* GROUP

Management Protocol Standards

- SNMP
- CMIP
- CIM/WBEM

Instrumentation API Standards

- ARM – Application Response Measurement
- AIC – Management objects in C
- JMX – Java management objects
- JSR 138 –
- JST 77 Application Server

Model and Schema Standards

- Applications
- QoS
- Policy
- Metrics

Architecture

- Management vs. Manageability
- Instrumentation

The DMTF Models

THE *Open* GROUP

- Applications model
 - Deployment
 - RunTime Application Management
- The Metrics Model
 - Unit of Work Metrics
 - Base Metrics
- The Policy Model
 - Conditions, Policys Actions

Objectives of App Management Model

THE *Open* GROUP

- Life-Cycle Management of Applications
 - Distribution, installation, etc.
- Management of running Applications, Services, Business Applications
 - Performance Information, Executing Operations, State of the application, etc.

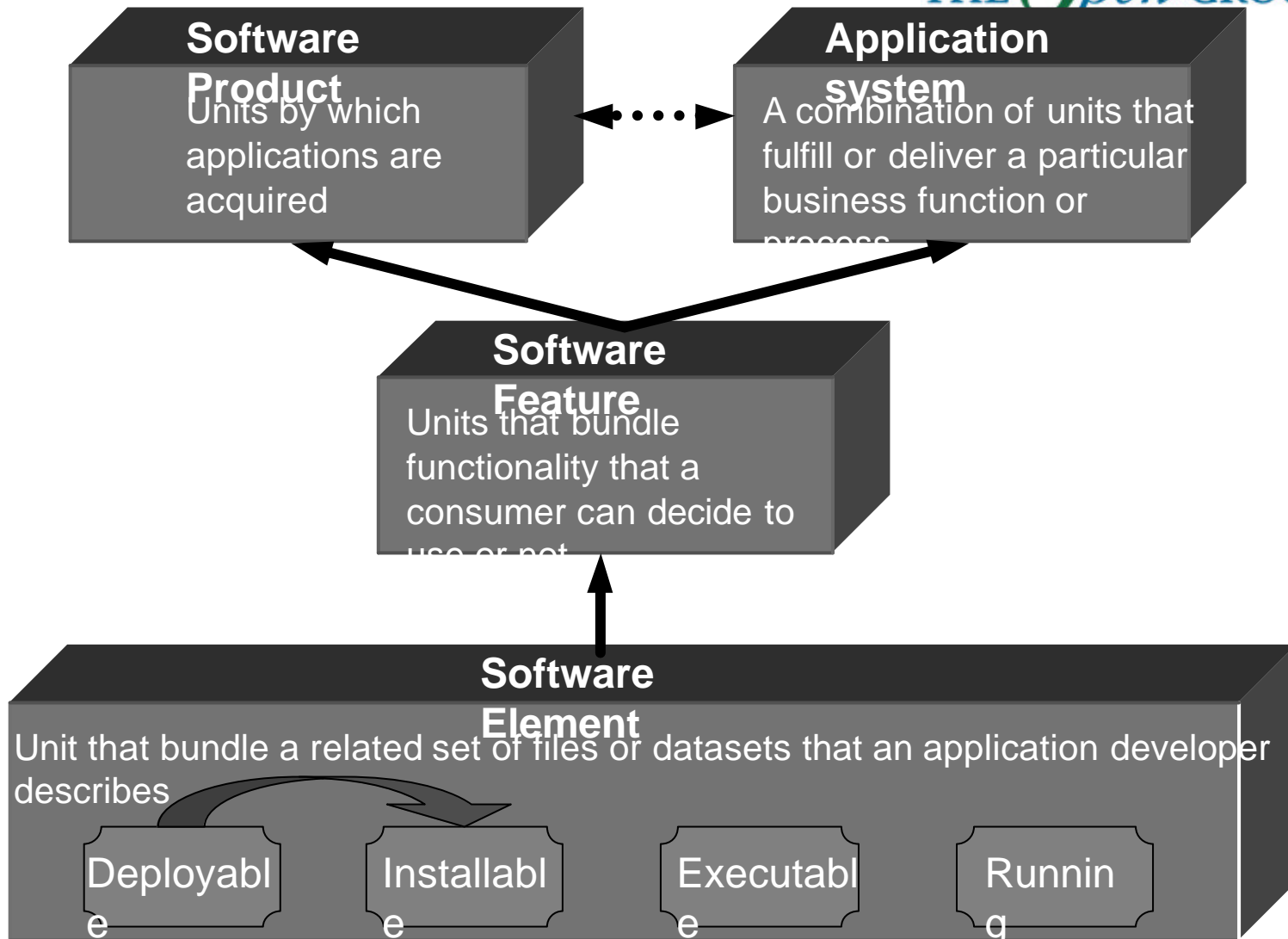
A Short History of the Group

THE *Open* GROUP

- Application Model – Lifecycle
 - One of the first models finished (1999)
- DAP -> Metrics
 - Model finished 2000
- Application RunTime
 - Group initiated 2000
 - First Submission CIM V2.5
 - Merge with DAP (common Metrics) 2001
 - Released Metrics extensions in May 2002

The Lifecycle Model

THE *Open* GROUP



Running Application Requirements

THE *Open* GROUP

Running Application Requirements have many interpretations.

They can focus on the “**Resource View**”...

↳ Model the operational issues for an application in isolation.

They can focus on the “**Service View**”...

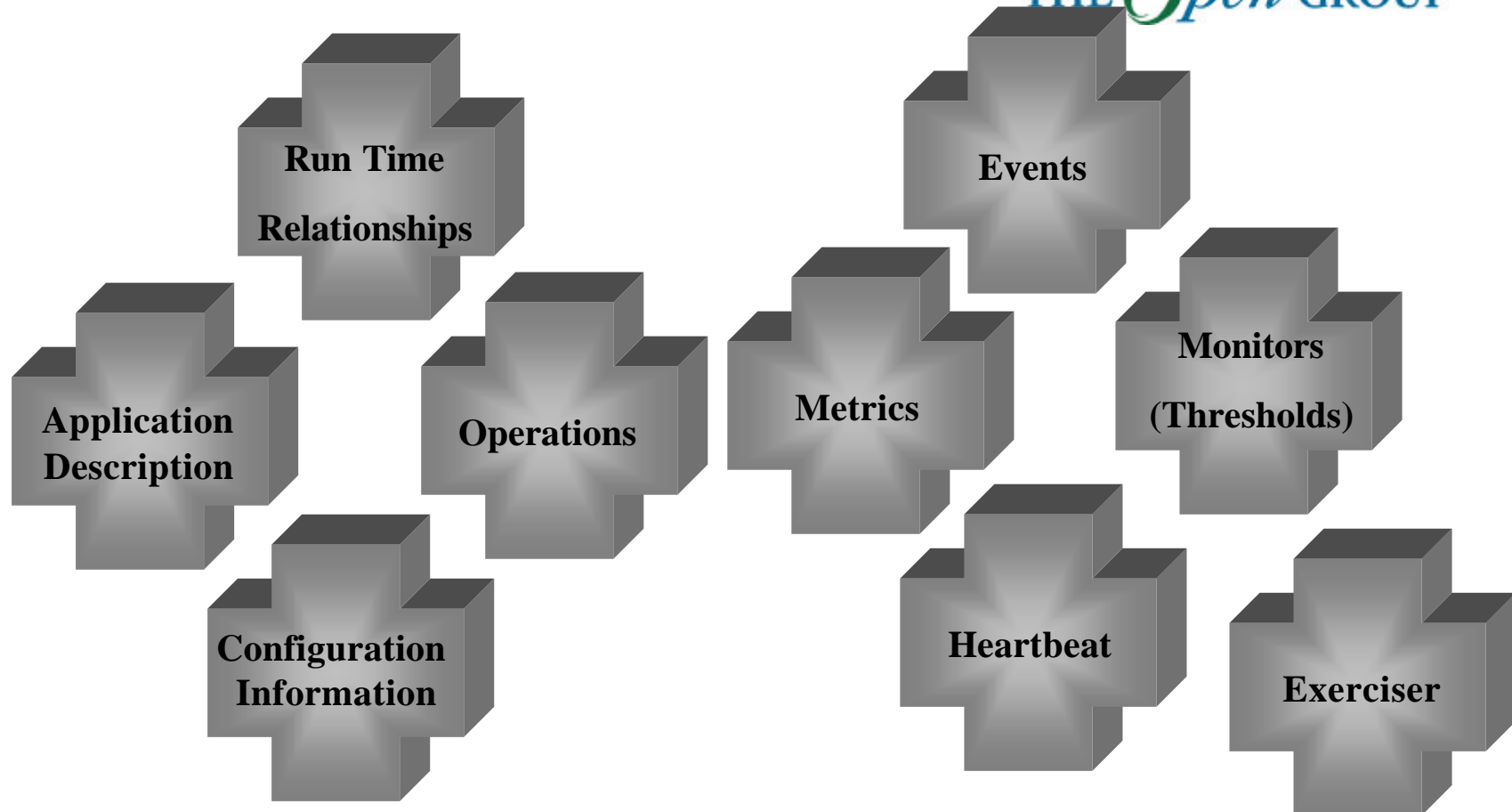
↳ Model the operational issues of a service for a user

They can focus on the “**Business View**”...

↳ Model the operational issues for an application that is part of a business process.

Running Application Ingredients

THE *Open* GROUP



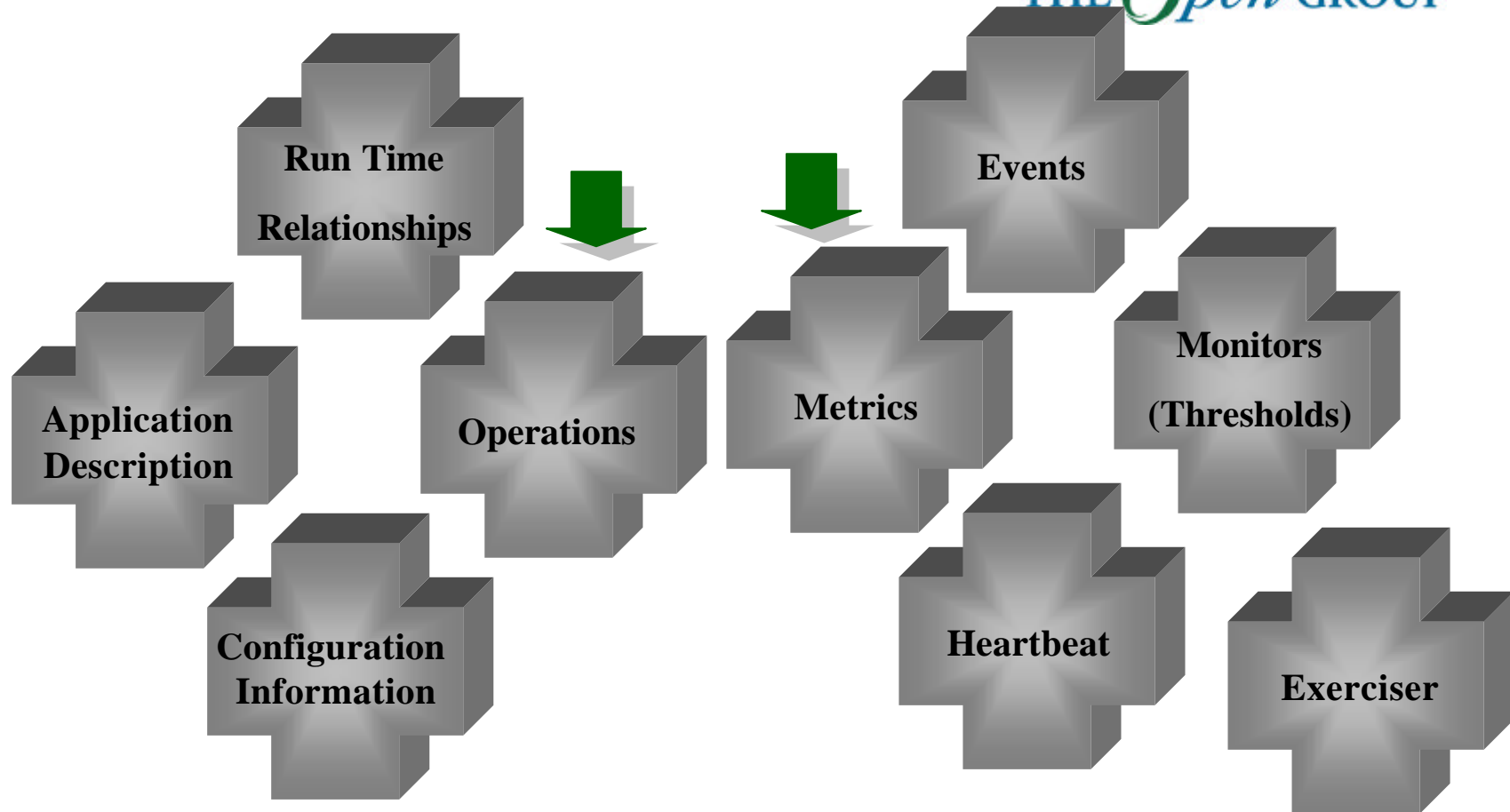
Current DMTF Work



- Metrics Model
 - Develop Prototypes for the Base Metrics
 - Expand the Metrics Model
 - Complete the base work on the Applications model
 - Operations
- Applications Model
 - Creating an Application RunTime model
 - Metrics
 - Operations
 - Application State

Running Application Ingredients

THE *Open* GROUP



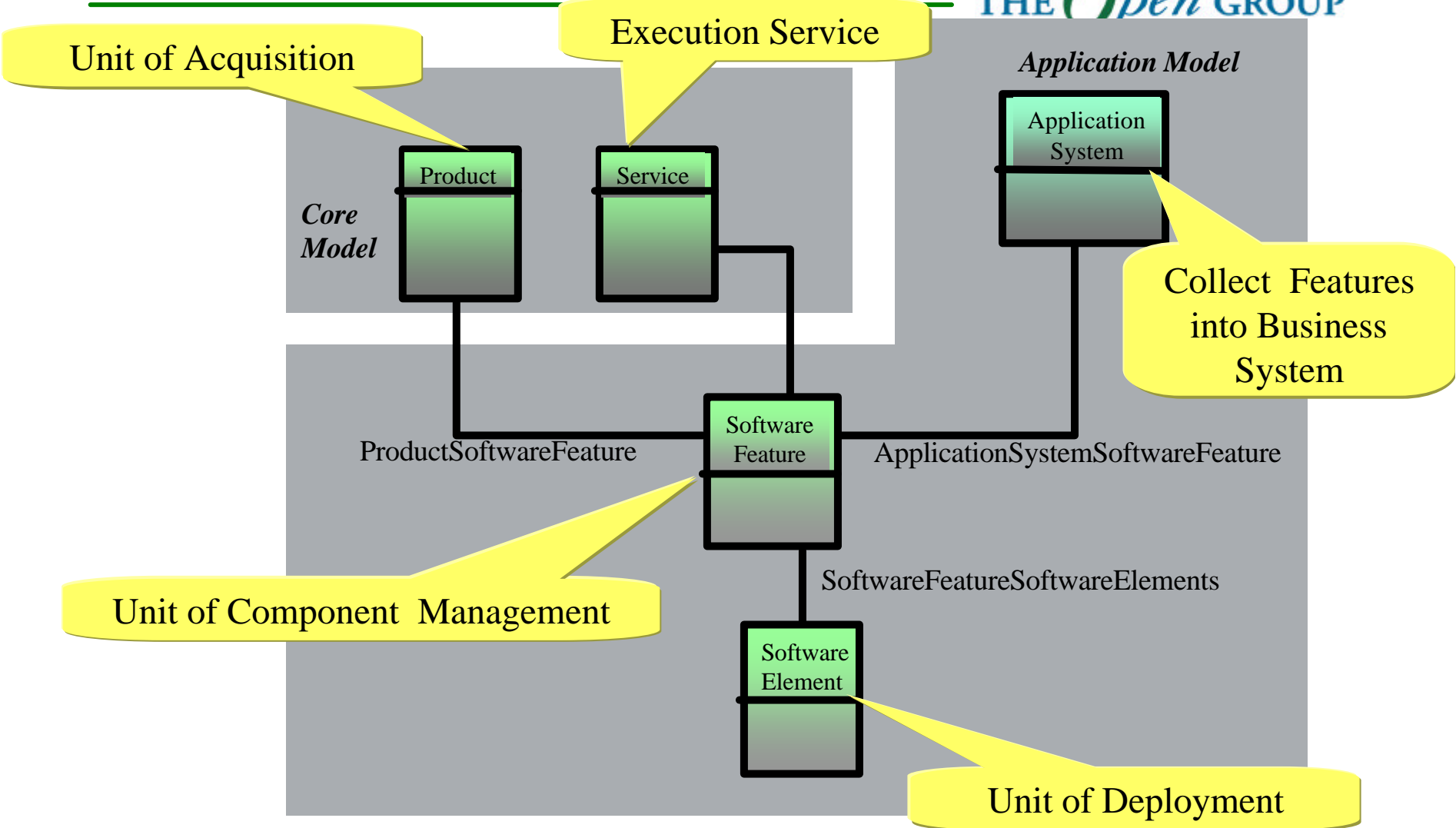
Life-Cycle Management

THE *Open* GROUP

Application Management Model
the
Application Life-Cycle

DMTF Lifecycle Model Overview

THE *Open* GROUP



Software Elements

THE *Open* GROUP

- ❑ The Unit of Deployment
 - Installed and uninstalled
- ❑ Represent a collection of files and associated details that are individually managed
- ❑ Abstracted to Software Features

This is what actually gets installed

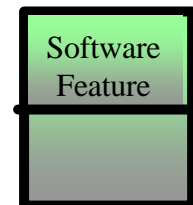


Software Feature

THE *Open* GROUP

- ❑ Unit of component management
- ❑ Components meaningful to consumer or user of the application
- ❑ Reflect functions or roles of a component of an application
- ❑ Ex. Trading client, quote server, etc.

This is what the user sees

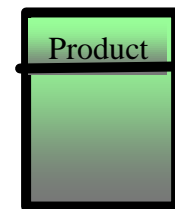


Software Product

THE *Open* GROUP

- ❑ Unit of acquisition
- ❑ Implies agreement between consumer and supplier
 - License, support, warranty, ...

This is what we pay for

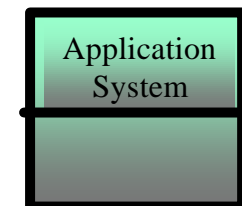


Application Systems

THE *Open* GROUP

- ❑ Collect features into business systems
- ❑ Support a business function
- ❑ Collections of software features from one or more products combined to fulfill business function

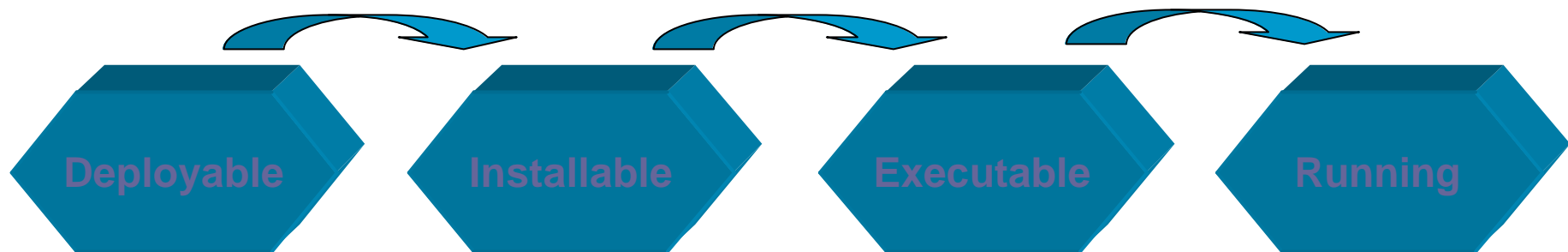
This is how the managers would like to view the environment



Application Life Cycle

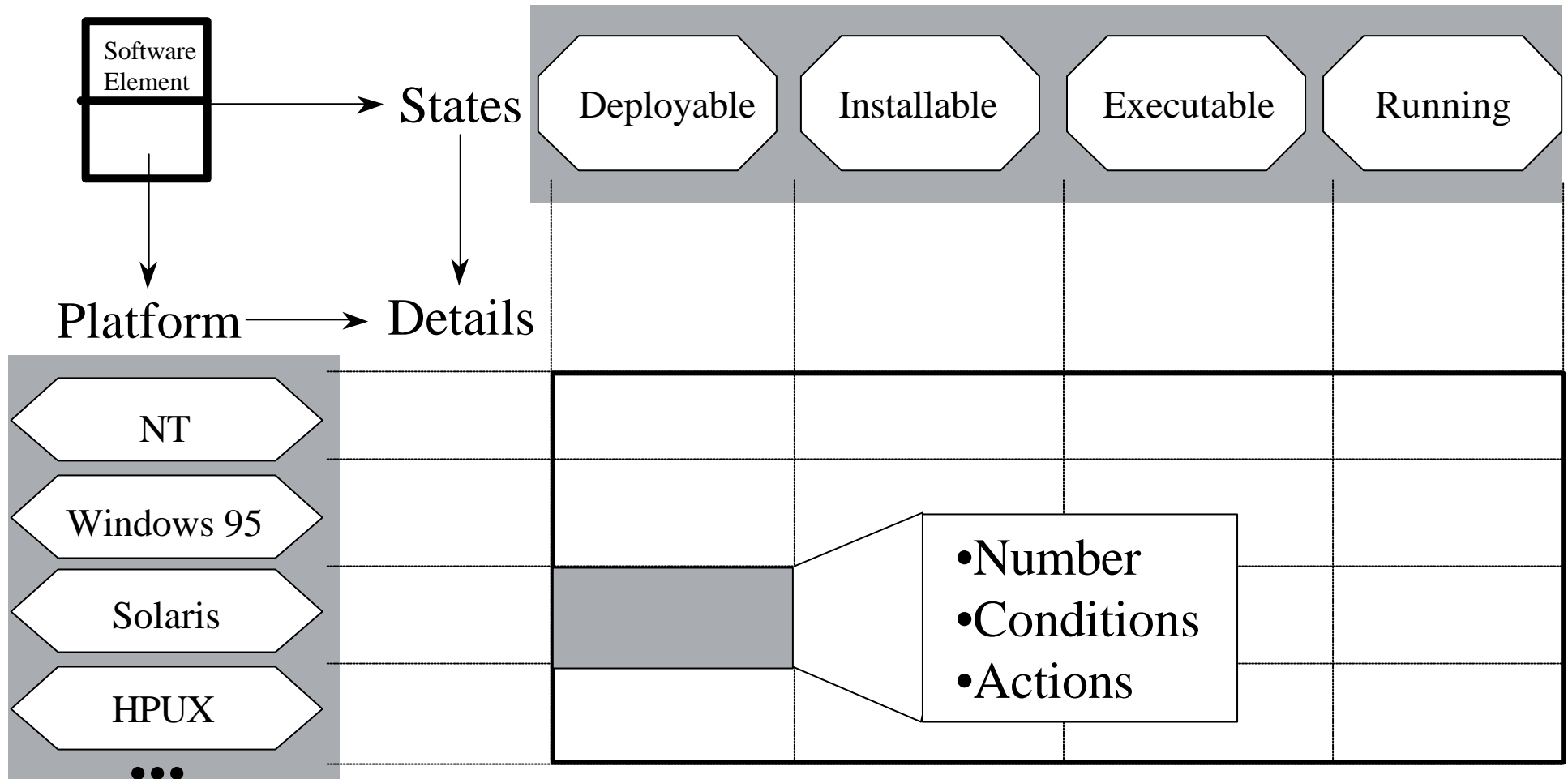
THE *Open* GROUP

- ❑ Critical states in process of transition from development to operational
- ❑ Applies to lowest-level component
 - Software Elements
- ❑ States



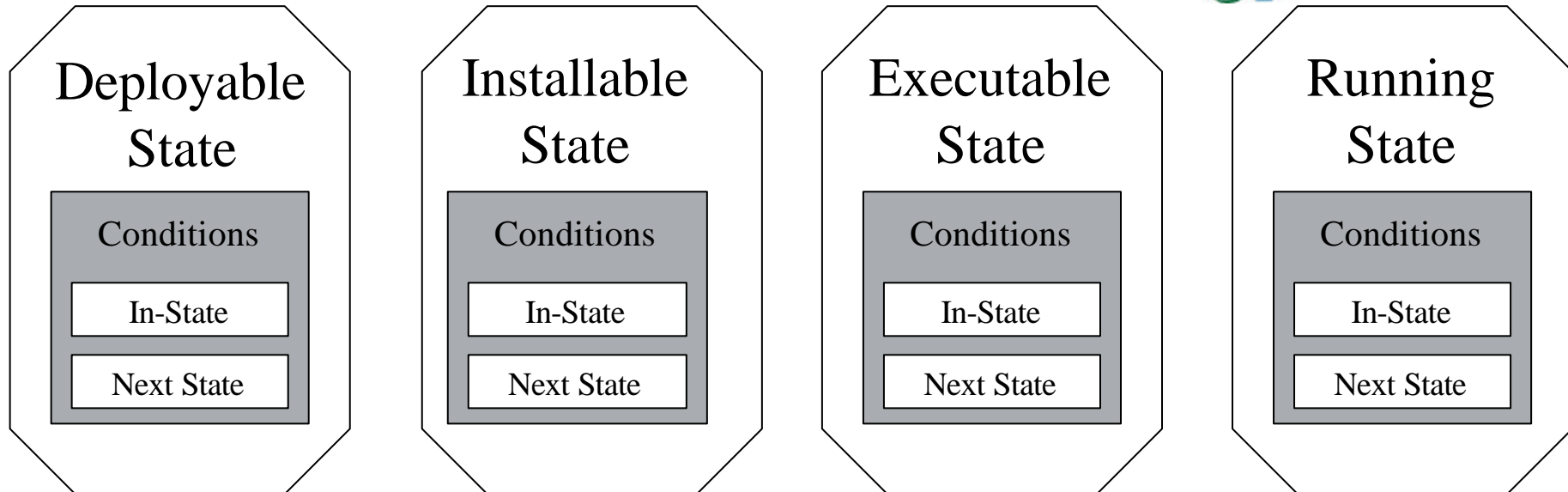
Refining Software Element

THE *Open* GROUP



Software Element Conditions

THE *Open* GROUP



Conditions are situations that are expected to exist or not exist in an environment

In-State Conditions are characteristics of an environment that contains an element.

Next-State Conditions are characteristics that need to be true in the target environment for the next state of a software element.

Software Element Conditions

THE *Open* GROUP

Condition	In-State Interpretation	Next-State Interpretation
Memory Requirements	Minimum Amount of memory required to transition into the <i>current</i> state.	Minimum amount of memory required to transition into the <i>next</i> state
Disk Space	Minimum amount of disk space required to transition into the <i>current</i> state.	Minimum amount of disk space required to transition into the <i>next</i> state.
Swap Space	Minimum amount of swap space required to transition into the <i>current</i> state.	Minimum amount of swap space required to transition into the <i>next</i> state.

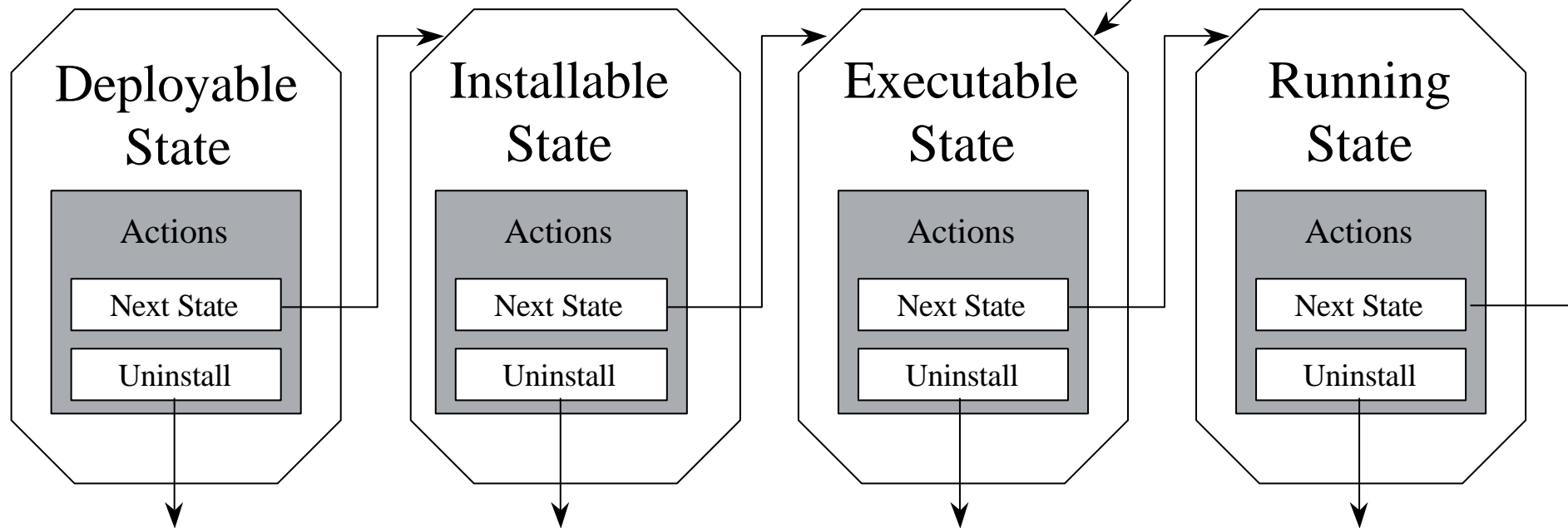
Software Element Conditions



Condition	In-State Interpretation	Next-State Interpretation
Architecture	The architecture required by a software element in the <i>current</i> state.	The architecture required by the software element to transition into the <i>next</i> state.
Files	A file that is expect to exist or not exist when a software element is in the <i>current</i> state.	A file this is expect to exist or not exist before a software element transitions into the <i>next</i> state.
Directories	A directory that is expect to exist or not exist when a software element is in the <i>current</i> state.	A directory this is expect to exist or not exist before a software element transitions into the <i>next</i> state.
OS Version	The version or ranges of versions a software element requires in its <i>current</i> state.	The version or ranges of versions a software element elements requires before it transitions into the <i>next</i> state.
Software Elements	A software element that is expect to exist or not exist when a software element is in the <i>current</i> state.	A software element that is expect to exist or not exist before a software element transitions into the <i>next</i> state.

Software Element Actions

THE *Open* GROUP



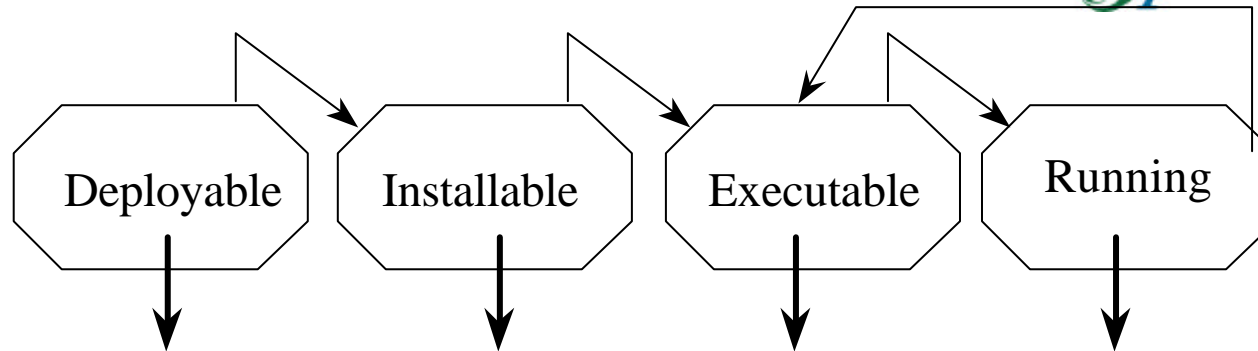
Actions are a sequence of operations

Next State Actions create a software element in a particular state.

Uninstall Actions properly remove a software element

Software Element Actions

THE *Open* GROUP



Actions	Description
Directory	An action to create or remove a directory.
File	An action to create or remove a file.
Re-boot	An action the signals the need to reboot the computer system.
Execute Program	An action that execute a program. This can be the install script or program (e.g., setup.exe) when a software element in the installable state transitions to the executable state.

**Application Management
Model**

The Metrics Model

-

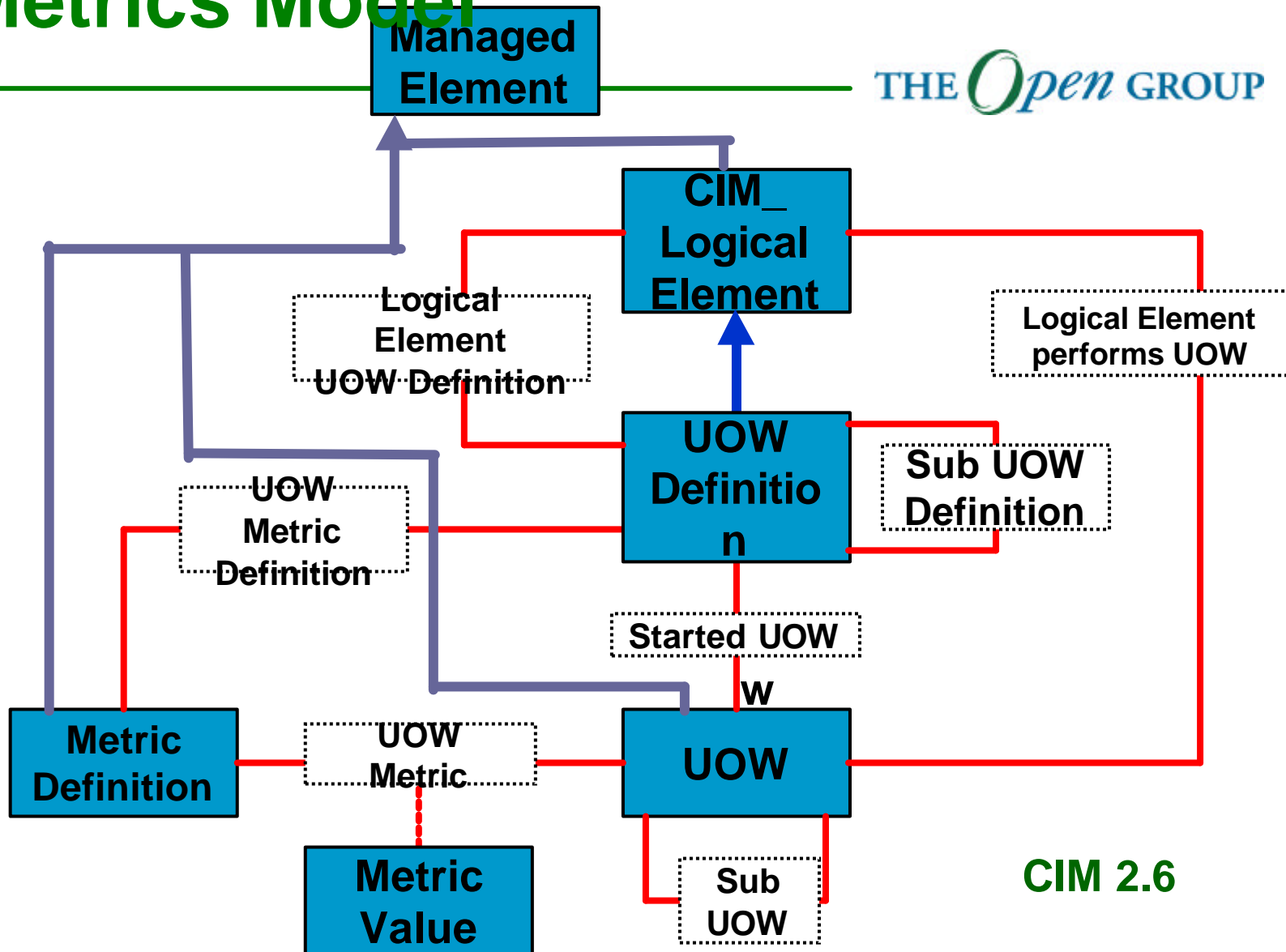
Measuring Performance

The Metrics Model

THE *Open* GROUP

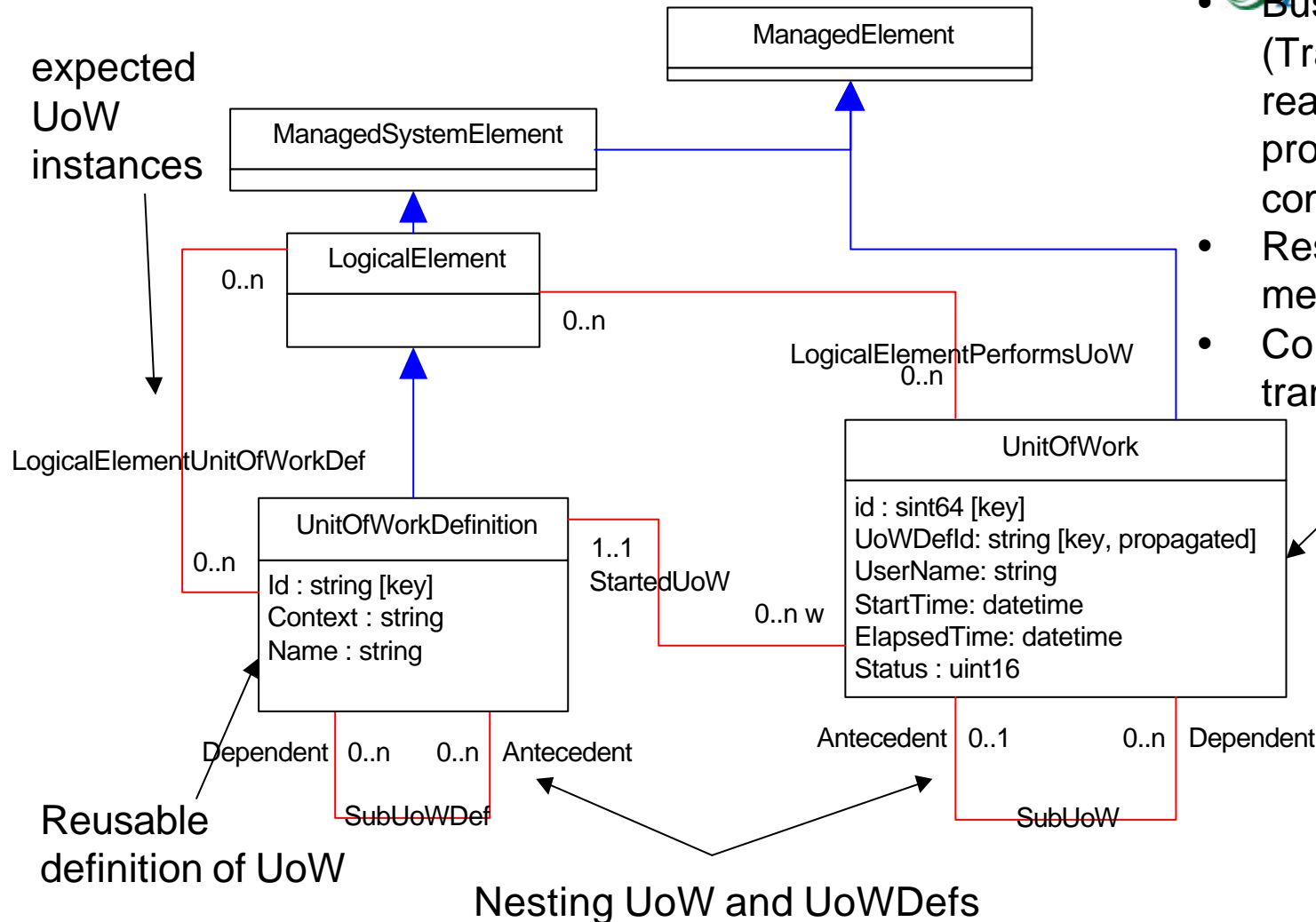
- Original Objective
 - Model for Performance Measurement (Unit of Work)
- Effective with Version 2.7 Preliminary
 - Model to capture and manipulate “general” metrics
 - And
 - Unit of work Metrics

Metrics Model



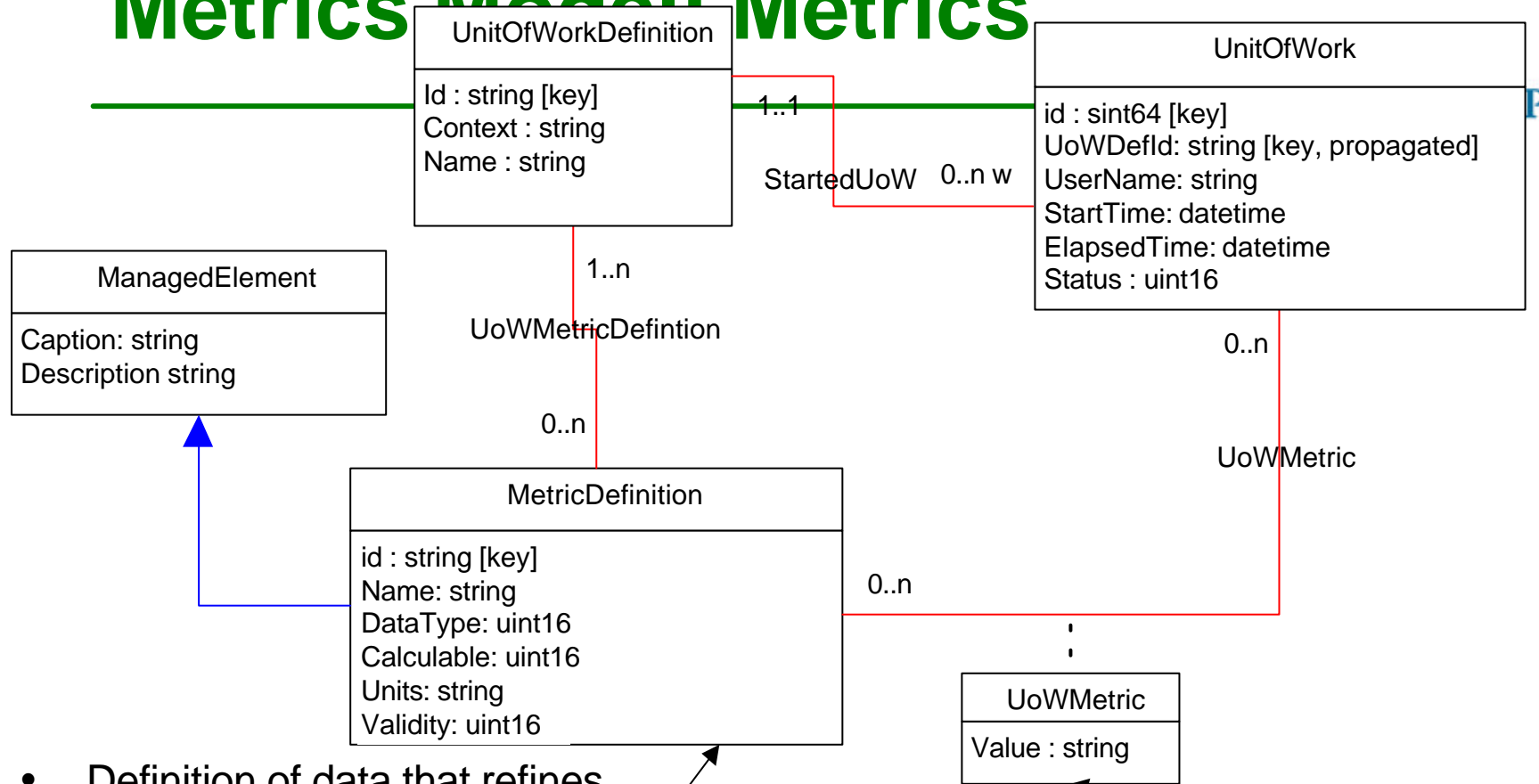
Metrics Schema: UnitOfWork

THE *Open* GROUP



- Business action (Transaction, Database read, Batchjob) in progress or recently completed
- Response time measurement
- Corresponds to ARM transactions

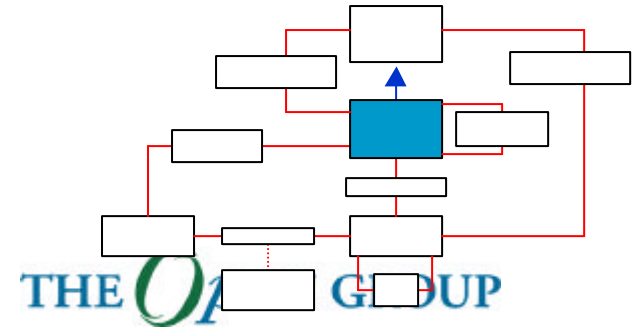
Metrics Model: Metrics



- Definition of data that refines UnitOfWork
- Reusable definition
- Counters, throughput, ...
- Only defined in the scope of UnitOfWorkDefinition

- current value of the metric
- generic due to type "string"

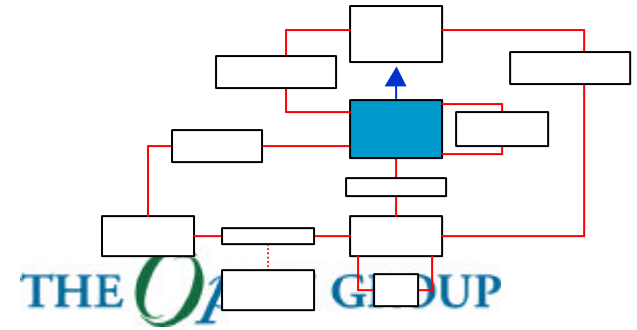
Unit Of Work Definition



- ❑ Defines a type of work
- ❑ Examples:
 - Update account balance
 - Backup file system
 - Query data server
 - Execute subroutine/method

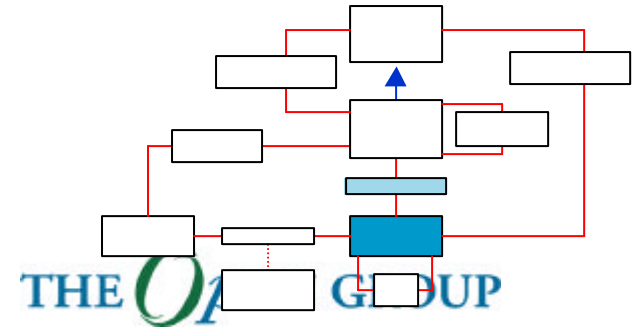
UOW Definition

Properties



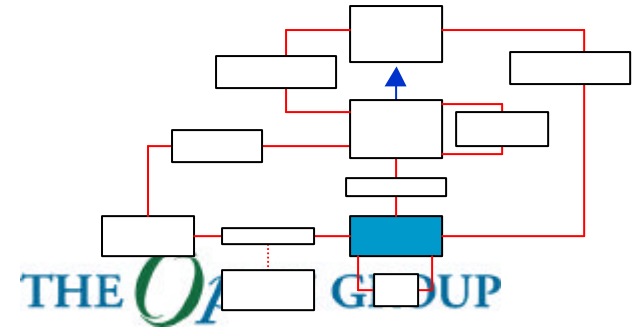
- ❑ ID - 16 bytes long [key]
 - Use of OSF UUID is recommended
 - MAC address (or substitute)
 - Current date and time
 - Counter to handle high-frequency allocations
 - A clock sequence and related state to handle the retrograde movement of clocks.
- ❑ Name
- ❑ Context (such as application name)

Unit of Work



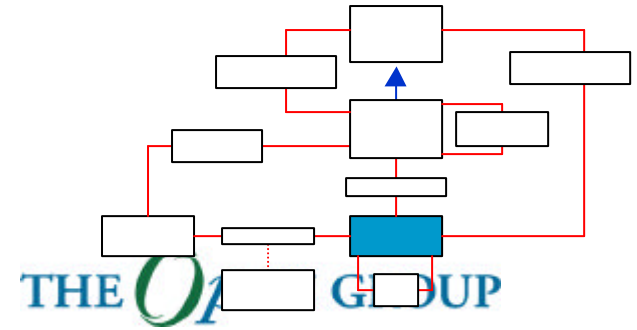
- ❑ Represents a UOW that has started (and maybe completed) executing
- ❑ Always associated to its definition
 - Weak association with propagated key
 - This provides separation of the namespace to avoid collisions

Unit of Work Measurements



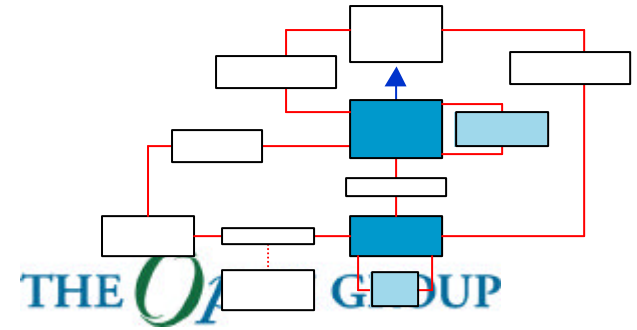
- ❑ Response time or elapsed time (if still executing)
- ❑ Status
 - Active
 - Suspended
 - Completed (with unknown state)
 - Completed Good
 - Completed Failed
 - Completed Aborted
- ❑ Metrics (in a separate class)

Unit of Work Properties



- ❑ ID [key]
- ❑ UOW ID [key, propagated]
- ❑ User Name [or ID or context or ???]
- ❑ Start time
- ❑ Elapsed time since UOW started (response time if complete)
- ❑ Status

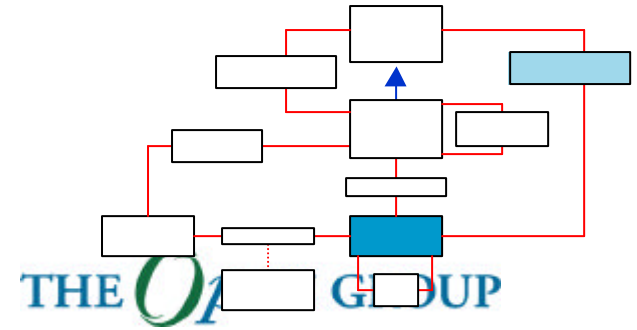
Associations between UOWs



- UOW may be associated to other units of work (parent or children)
 - Ex: a server UOW may be the child of the client UOW that invoked it
- Association can be at either or both of two levels
 - Between UOW definitions
 - Between UOW instances

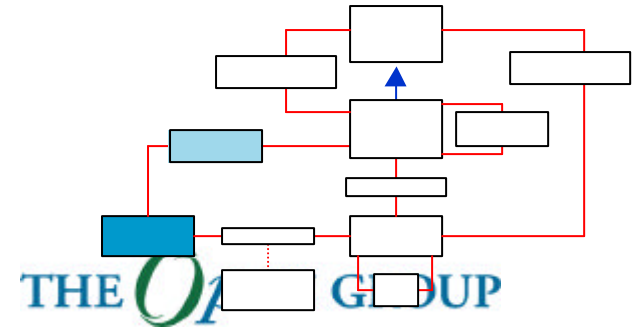
Associations to

Logical Elements



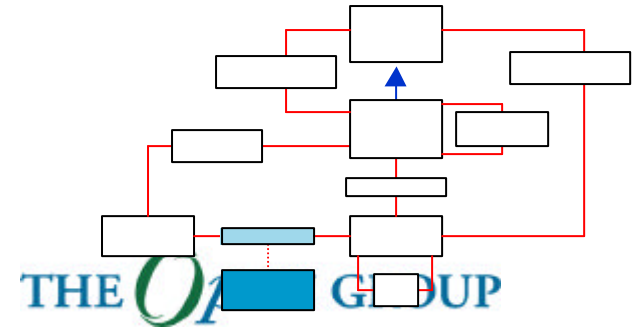
- ❑ A UOW instance may be associated to the Logical Element that performs it
- ❑ Examples:
 - Computer system/processor
 - Printer
 - Application system
- ❑ A UOW definition may be associated to another logical element
- ❑ Semantics of the association are not specified

Metric Definition



- The Class
 - Defines metrics associated with a unit of work
 - Even if defined, a value may not be provided at all times
- Properties
 - ID [key]
 - Name
 - Data type (sint32, uint64, string, ...)
 - Calculable (What type of calculations, if any, are OK: none, sum, no sum)
 - Units (bytes, files, milliseconds, ...)
 - Validity (start/middle/stop) [??]

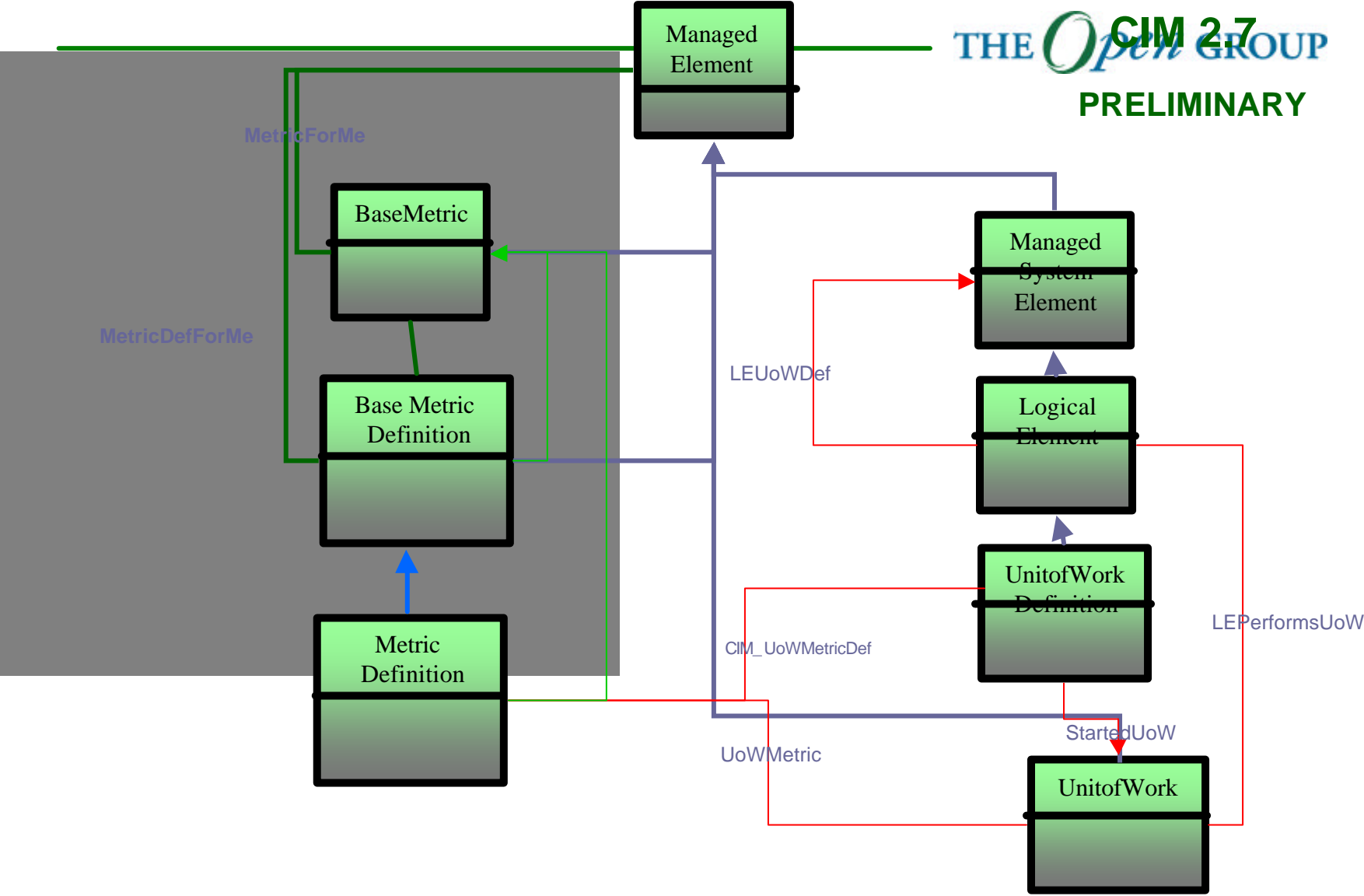
Metric



- ❑ Contains the actual value at any point in time
- ❑ Value is a string so it can contain any of the several data types
- ❑ Constraint on Metric Definition:
 - Must be associated to the same UOW Definition to which the UOW is weakly associated

The Metrics Model

THE *Open* GROUP
CIM 2.7
PRELIMINARY



Base Metric Definition

- The Class
 - Defines metric associated with a ManagedElement
 - Even if defined, a value may not be provided at all times
- Properties
 - ID [key]
 - Name
 - Data type (sint32, uint64, string, ...)
 - Calculable (What type of calculations, if any, are OK: none, sum, no sum)
 - Units (bytes, files)

BaseMetricDefinition
Id : string {key}
Name: string
DataType: uint16
Calculable: uint16
Units: string
BreakdownDimensions: string[]

Base Metric

- ❑ Contains the actual value at any point in time
- ❑ Value is a string so it can contain any of the several data types
- ❑ Properties to add general semantics to metrics
 - BreakdownDimension
 - BreakdownValue
 - Volatile
 - Duration

BaseMetricValue
InstanceId : string {key}
MetricDefinitionId: string {required}
MeasuredElementName: string
TimeStamp: datetime
Duration: datetime
MetricValue: string
BreakdownDimension: string
BreakdownValue: string
Volatile: boolean

UofW Changes

- ❑ Trace Level
 - Allows adding information to UofW Definition to support tracing of transactions through the system

TraceLevelType
Instanceld : string {key}
BaseType : uint16
Values: uint32[] {required}
ValueDescriptions: string[] {required}

Application Management Model

Looking Ahead

-

Objectives, Strategies and
new work

Modeling, the next generation

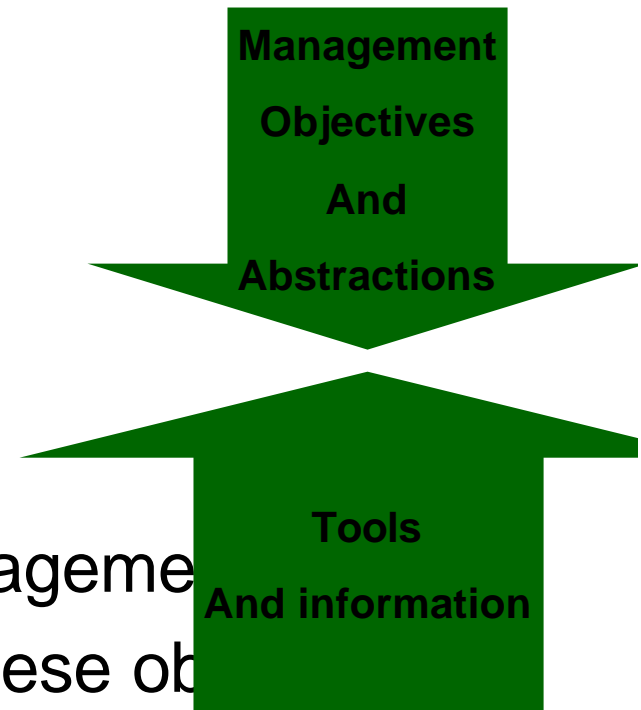
THE *Open* GROUP

- ❑ Expanding the model of the Applications
- ❑ Growing Metrics
- ❑ Completing the Application Runtime Model
- ❑ Closing the loop
 - Autonomic computing, etc.
 - Important model characteristics
 - Measurement
 - Analysis
 - Affect
- ❑ The Modeling components
 - QoS
 - Policy
 - Managing the Application
 - Metrics, Operations, State

Expanding the Application Model

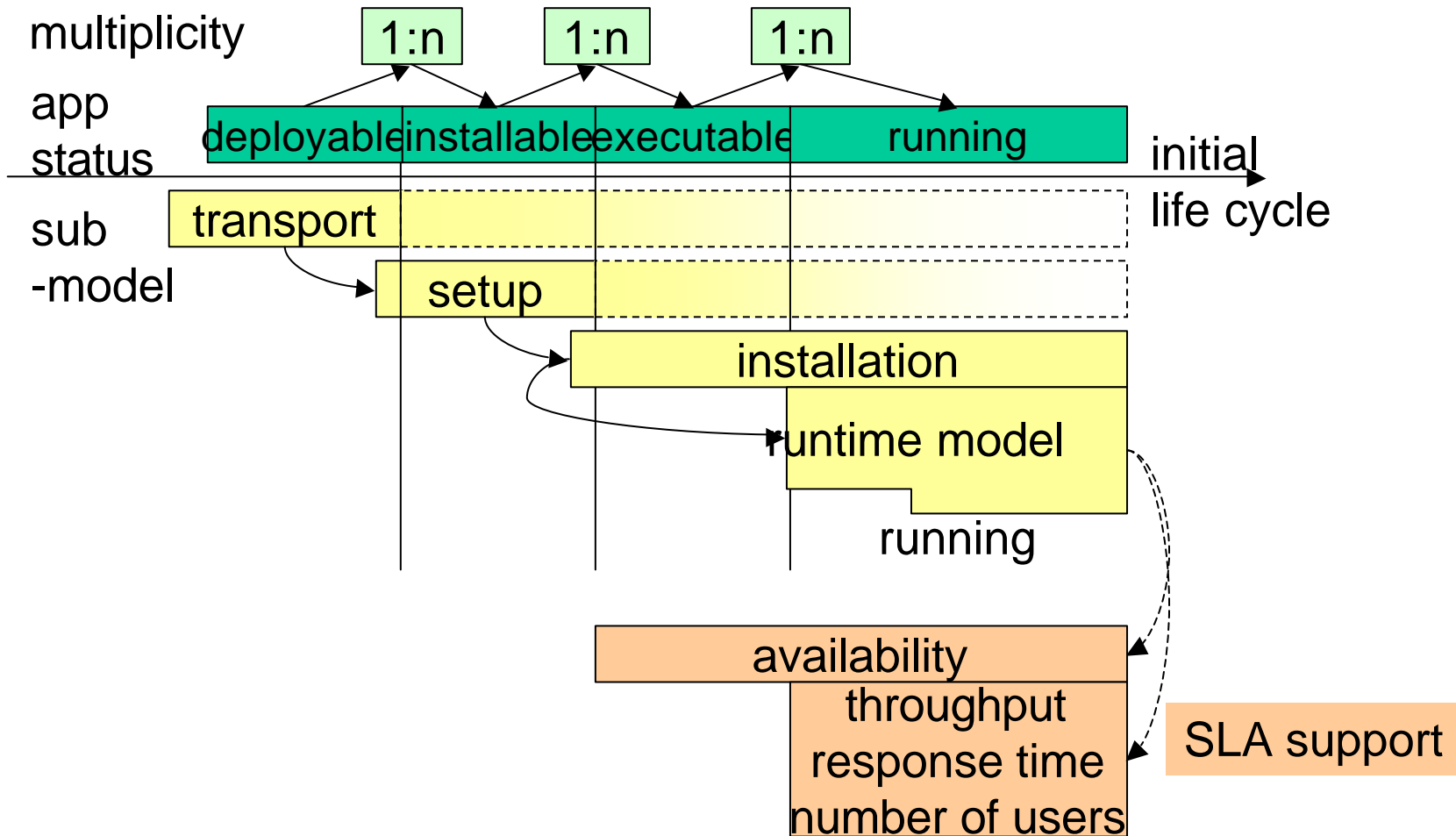
THE *Open* GROUP

- ❑ Expanding the runtime “tools”
 - Metrics
 - Operations
 - State
- ❑ Expanding the runtime model
 - Top Down – Objectives of Management
 - Expand the model to support these objectives
 - Manageability is not management
 - But Without management it is nothing.
- ❑ The lifecycle model
 - Aren't checks and actions really policy?

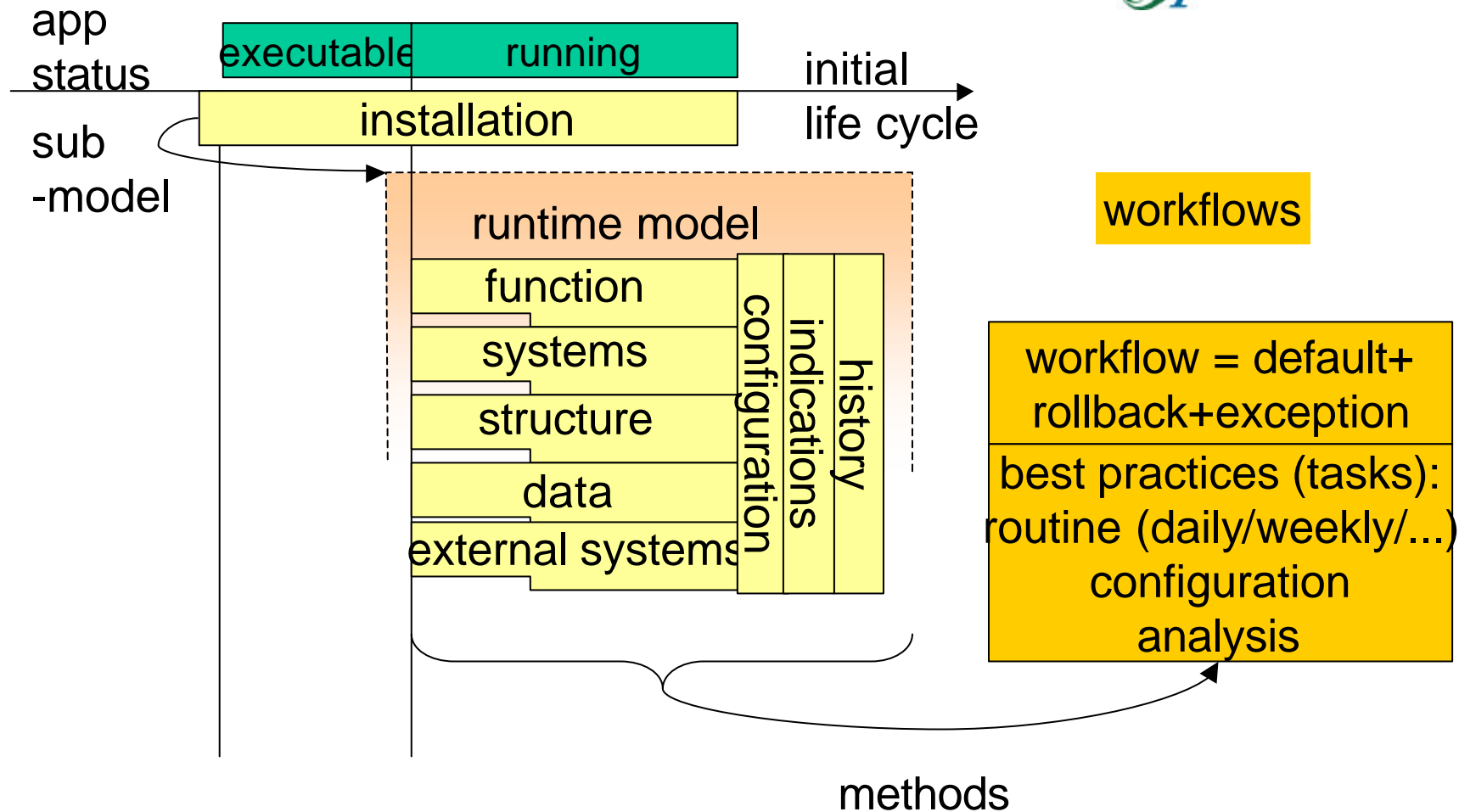


First Layer of Sub-Models

THE *Open* GROUP

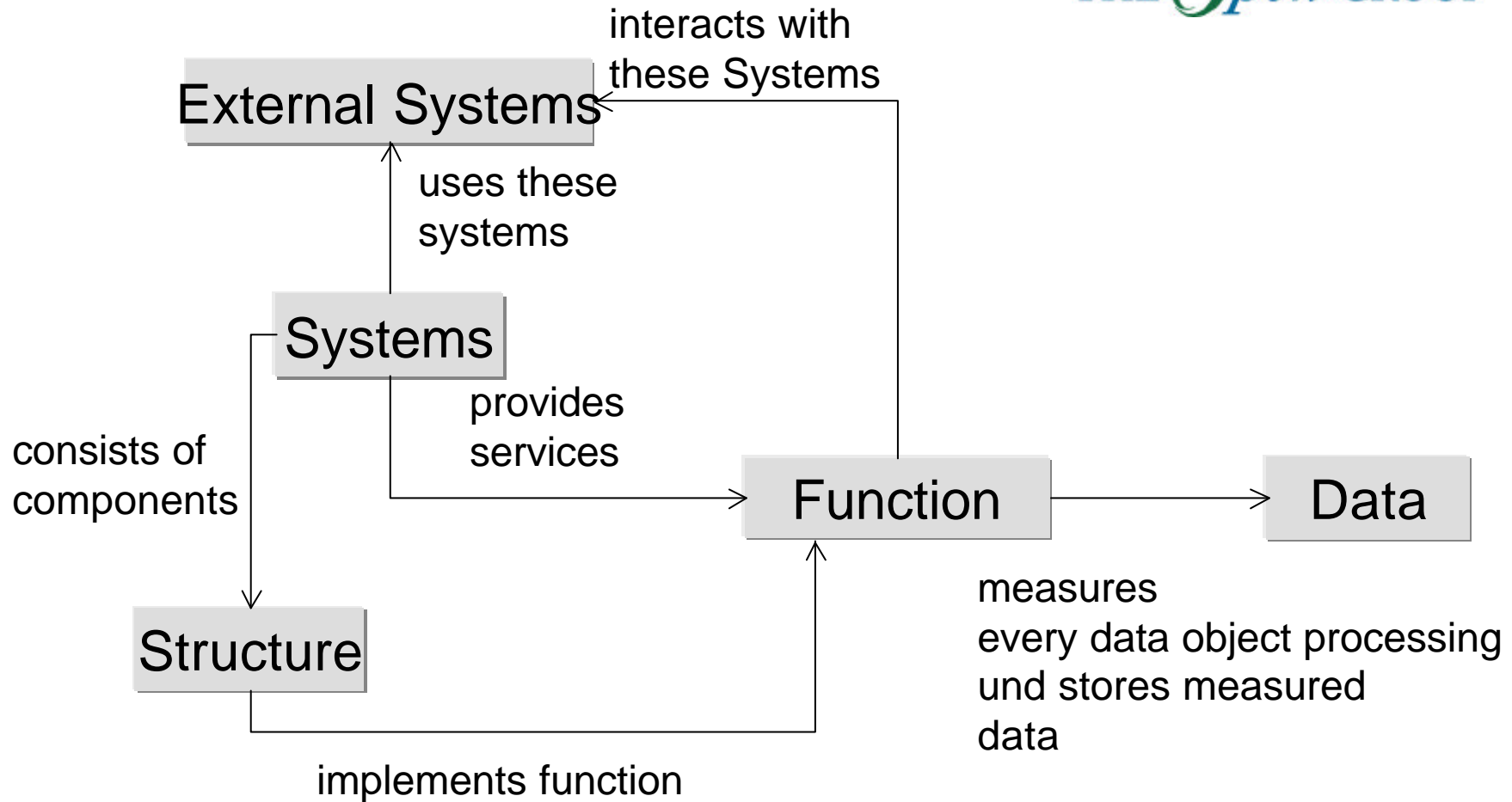


Runtime Model



Runtime Sub-Model Relationships

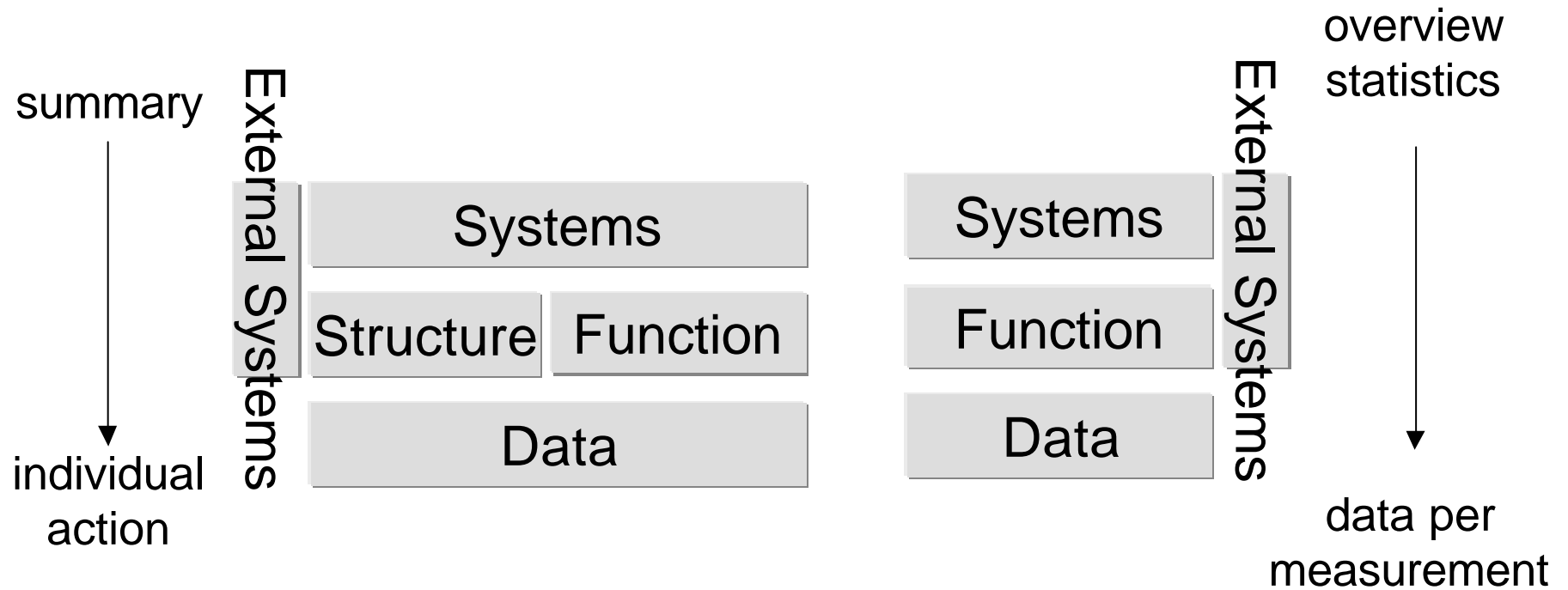
THE *Open* GROUP



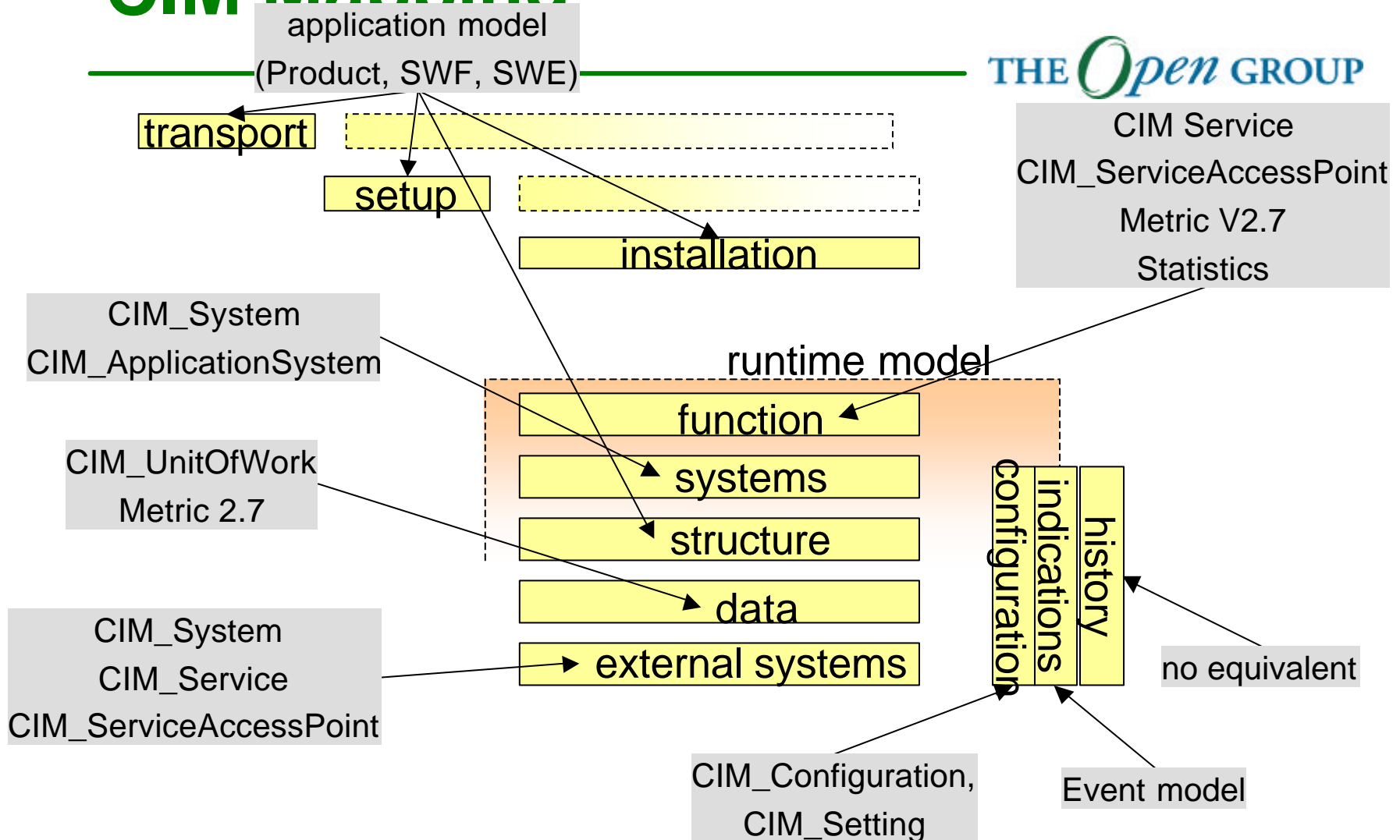
Levels of Detail

fault (errors), status

performance



CIM Mapping



Conclusions

THE *Open* GROUP

- ❑ Work in Process today
- ❑ Joint group but with no real TOG participation
- ❑ Needs user participation
- ❑ Needs another shot of requirements, etc.