



ACS-WG
Installable Unit Deployment
Descriptor

GGF13 , March 14, 2005

Thomas Studwell
studwell@us.ibm.com

IBM Autonomic Computing

<http://forge.gridforum.org/projects/acs-wg/>



Intellectual Property Policy

- I acknowledge that participation in GGF13 is subject to the GGF Intellectual Property Policy.
- **Intellectual Property Notices Note Well:** All statements related to the activities of the GGF and addressed to the GGF are subject to all provisions of Section 17 of GFD-C.1 (.pdf), which grants to the GGF and its participants certain licenses and rights in such statements. Such statements include verbal statements in GGF meetings, as well as written and electronic communications made at any time or place, which are addressed to: the GGF plenary session, any GGF working group or portion thereof, the GFSG, or any member thereof on behalf of the GFSG, the GFAC, or any member thereof on behalf of the GFAC, any GGF mailing list, including any working group or research group list, or any other list functioning under GGF auspices, the GFD Editor or the GWD process
- Statements made outside of a GGF meeting, mailing list or other function, that are clearly not intended to be input to an GGF activity, group or function, are not subject to these provisions.
- Excerpt from Section 17 of GFD-C.1 Where the GFSG knows of rights, or claimed rights, the GGF secretariat shall attempt to obtain from the claimant of such rights, a written assurance that upon approval by the GFSG of the relevant GGF document(s), any party will be able to obtain the right to implement, use and distribute the technology or works when implementing, using or distributing technology based upon the specific specification(s) under openly specified, reasonable, non-discriminatory terms. The working group or research group proposing the use of the technology with respect to which the proprietary rights are claimed may assist the GGF secretariat in this effort. The results of this procedure shall not affect advancement of document, except that the GFSG may defer approval where a delay may facilitate the obtaining of such assurances. The results will, however, be recorded by the GGF Secretariat, and made available. The GFSG may also direct that a summary of the results be included in any GFD published containing the specification.

GGF Intellectual Property Policies are adapted from the IETF Intellectual Property Policies that support the Internet Standards Process.



- **The Problem Space**
- **Principles**
- **The Installable Unit Deployment Descriptor Relationship to ACS**
- **IUDD, what it is and what it isn't...**
- **IUDD concepts, structure and capabilities**
- **Questions**
- **Summary**



- **Installation and configuration complexities must not become an impediment to adoption of grid technologies**
- **Installation and configuration needs are fundamental – little is unique to grid**
- **Grid, by definition, requires automation**
 - **Static configurations, tolerated in data center, can not be tolerated in grid**
 - **Standardized deployment descriptors increase reliability**
- **What standards can provide a foundation?**



Principles

- For more complete autonomic functionality, the installation of OS and grid container must be automated and born from the network.
- Generic enough to apply to any computing container solution. Grid or otherwise.
- Must be able to deal with heterogeneous pools of hardware.
- An increasing percentage of software is aggregated as a component within a larger, integrated “solution”. Grid applications are, by definition, an aggregated solution
- Customers outages are often caused by their inability to rollout changes to applications because of the complex interdependencies with other application components and products
- In order to enable autonomic deployment and configuration management , standardized formats are needed for declaring the structure of a solution and dependencies among its software components
 - Grid depends on the ability to dynamically deploy and configure solutions



- **Application Contents Service defines a repository interface (ARI) and format for contents of the repository (AAF)**
- **While the requirements for a grid application archive are unique to grid, the description of the contents are not**
- **The description must define the application artifacts, dependencies, and deployment mechanisms**
- **Add software life cycle management to the mix and you have the requirements for Installable Unit Deployment Descriptor (IUDD)**
- **Only requirements difference between AAF and IUDD are any wrappers needed for storage within ACS repository.**



- **IBM, InstallShield (Macrovision), Zero G, and Novell collaborated on a set of specifications to**
 - **“define the schema of an XML document describing the characteristics of an installable unit (IU) of software that are relevant for its deployment, configuration and maintenance. ”**
 - **Published by W3C on July 15, 2004**
 - <http://www.w3.org/Submission/2004/04/>
 - **Announcement:** <http://xml.coverpages.org/ni2004-07-19-a.html>
 - **Made available to Industry under RF terms**
 - **Publication coincident with announcements calling for formation of a standards workgroup to formalize an Industry standard for IUDD Schema.**



IUDD – What it is... (cont.)

■ **Scope**

- **Atomic installable units as small as mobile devices**
- **to...**
- **Enterprise scale applications that include services distributed over a Grid.**
 - **Designed for distributed heterogeneous environments**
 - **See Application Contents Service BoF**

■ **Standardization goal**

- **Have a single Industry standard to describe all aspects of a software solution needed to provide complete lifecycle maintenance.**
 - **IUDD specifications will be IBM's submission to a formal workgroup.**



IUDD – What it isn't...

- **Does NOT define the**
 - **Hosting platform information models**
 - DMTF is responsible for these
 - » (with input from other orgs like GGF)
 - **Hosting platform management interfaces**
 - OASIS WS-Distributed Management defining for WS
 - » GGF CMM-WG provides Grid specific extensions
 - **Deployment and Lifecycle Management engines**
 - Typically proprietary or value-added
 - » For example, see InstallShield and Zero G announcements
 - Open platforms, e.g. GGF, are defined by respective orgs

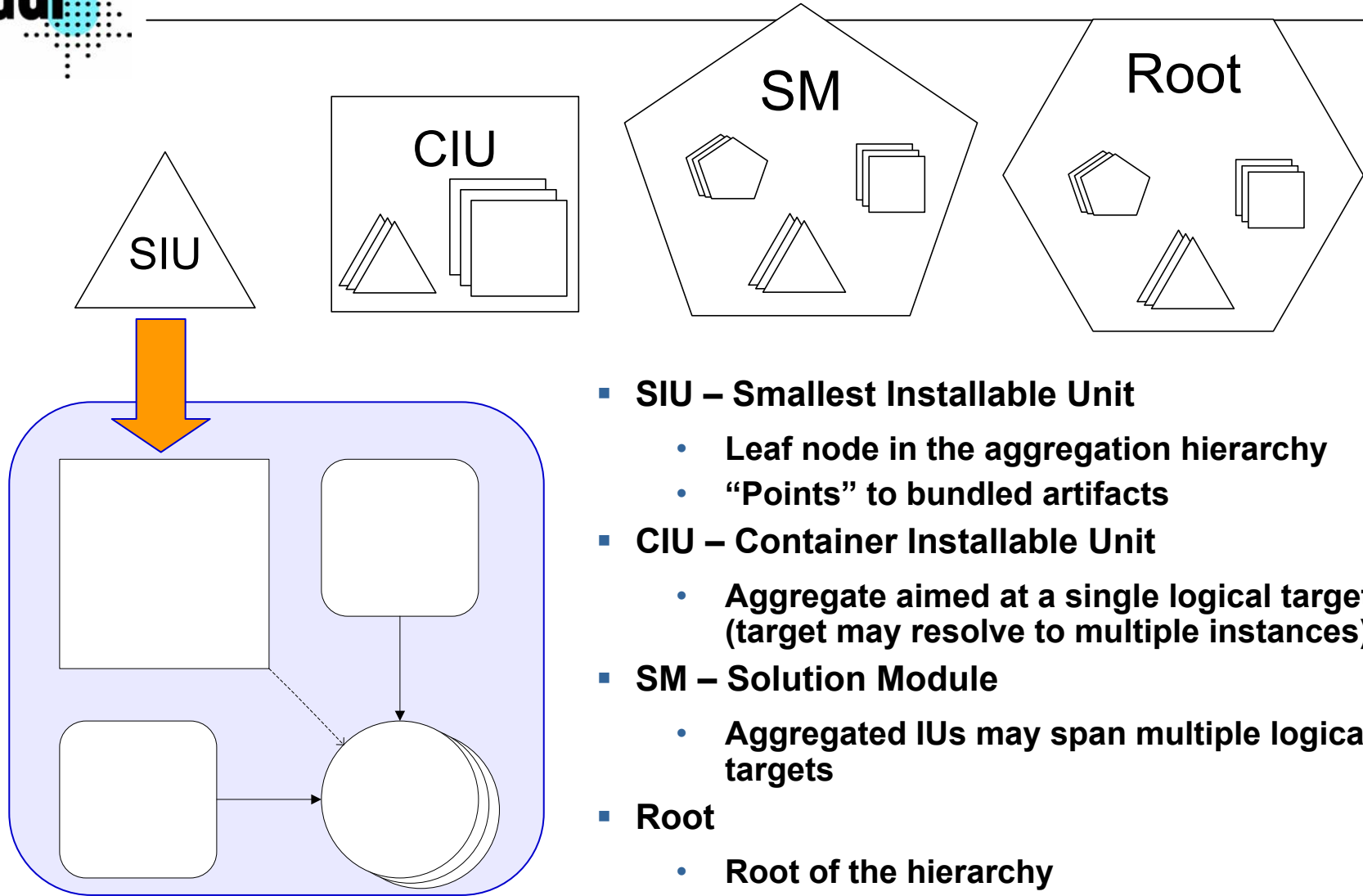


Installable Unit Deployment Descriptor - Concepts

- **A solution may encompass inter-dependent Installable Units (IU) deployed across multiple hosting domains**
- **Target hosting domains (Op Systems, J2EE servers, Databases) and other external resources are each represented by a node in the solution topology.**
- **The simplest solution has a single IU targeted at the only topology node.**
Example: AcrobatReader targeted to a Windows OS
- **A topology node needs to be mapped to one or more instances of a resource (e.g. OS).**
The mapping is constrained by *type* and possibly additional *selection requirements*.
- **Installable Units declare dependencies** (topology targets may also declare dependencies)
- **Requirements from all IU's aimed to a given topology target include:**
 - hosting-environment **dependencies** (capacity, consumption, properties, relationships, etc)
 - software **dependencies** (pre- co- and ex-requisites)
- **A solution may define *bundled requisites* that can be deployed on-demand to satisfy a software dependency**



Installable Units

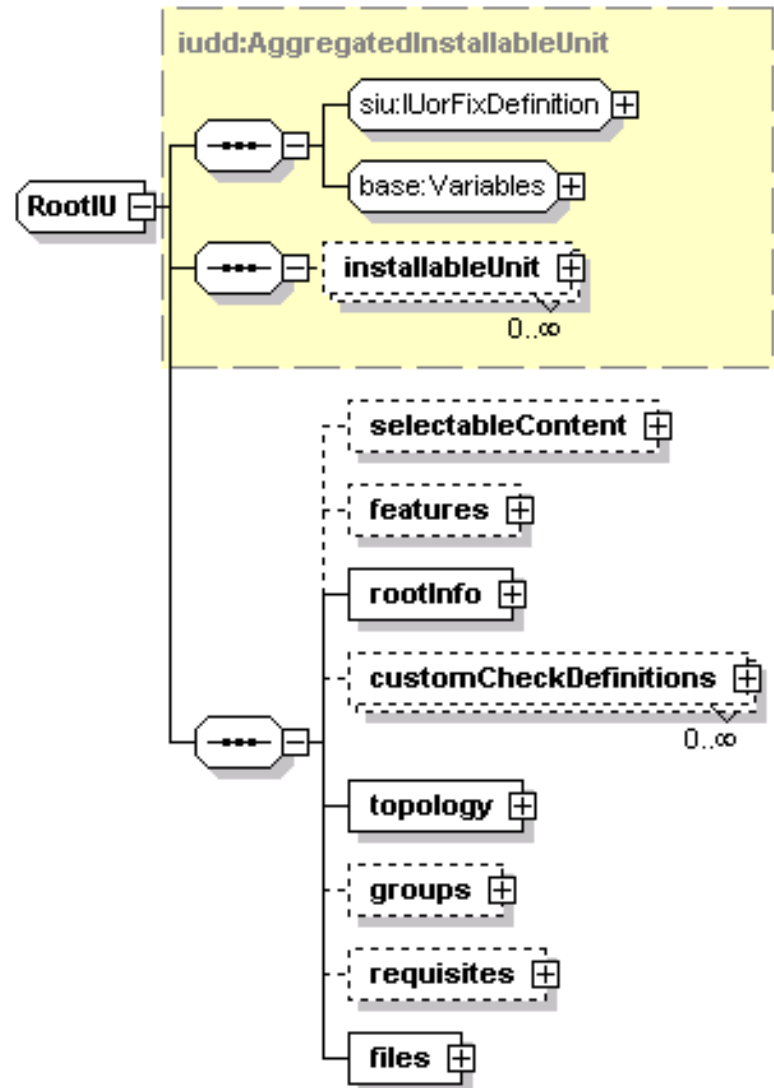


- **SIU – Smallest Installable Unit**
 - Leaf node in the aggregation hierarchy
 - “Points” to bundled artifacts
- **CIU – Container Installable Unit**
 - Aggregate aimed at a single logical target (target may resolve to multiple instances)
- **SM – Solution Module**
 - Aggregated IUs may span multiple logical targets
- **Root**
 - Root of the hierarchy
 - Unit of packaging



IUDD – Root IU

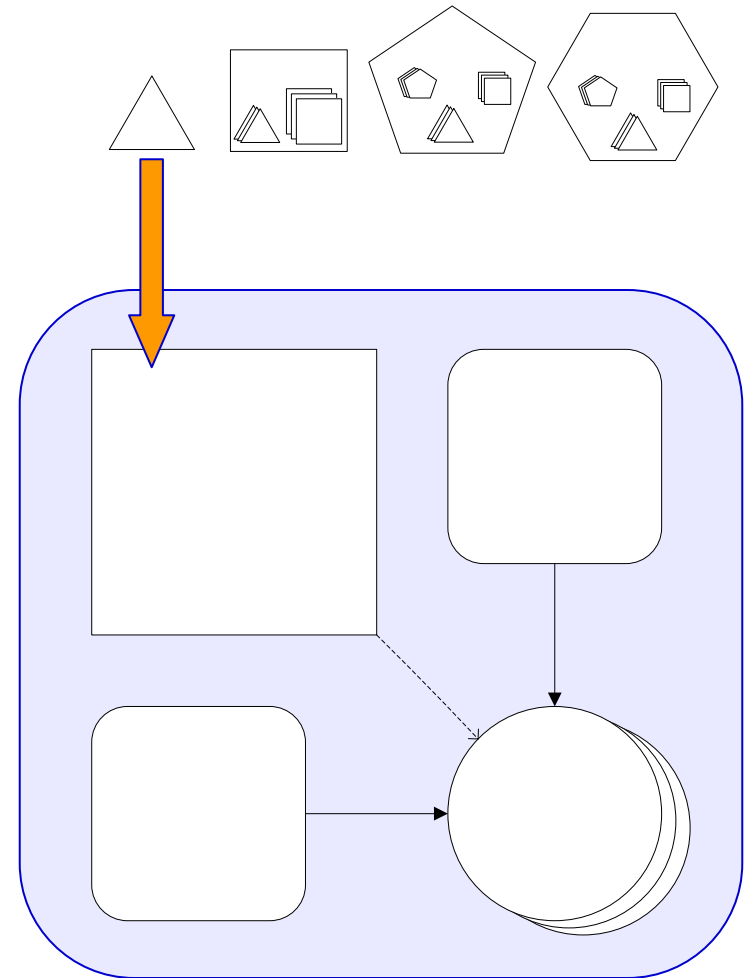
- **Root IU is the Unit of packaging**
 - It is an Installable Unit
 - Base, Update or Fix
 - May include any other IU aggregate
- **Base and selectable content**
- **Topology**
 - Target definitions
- **Info**
 - Build #
 - Manufacturer
 - Size
- **Features and Groups**
 - Features select optional content
 - Groups are pre-defined sets of features
- **Bundled requisites**
- **Files** (included in the package)
 - Files referenced from within the IUDD
 - Artifacts
 - Other bundled root IU packages (IUDD)
 - Files referenced from within artifacts





Targeting, artifacts, bundled files

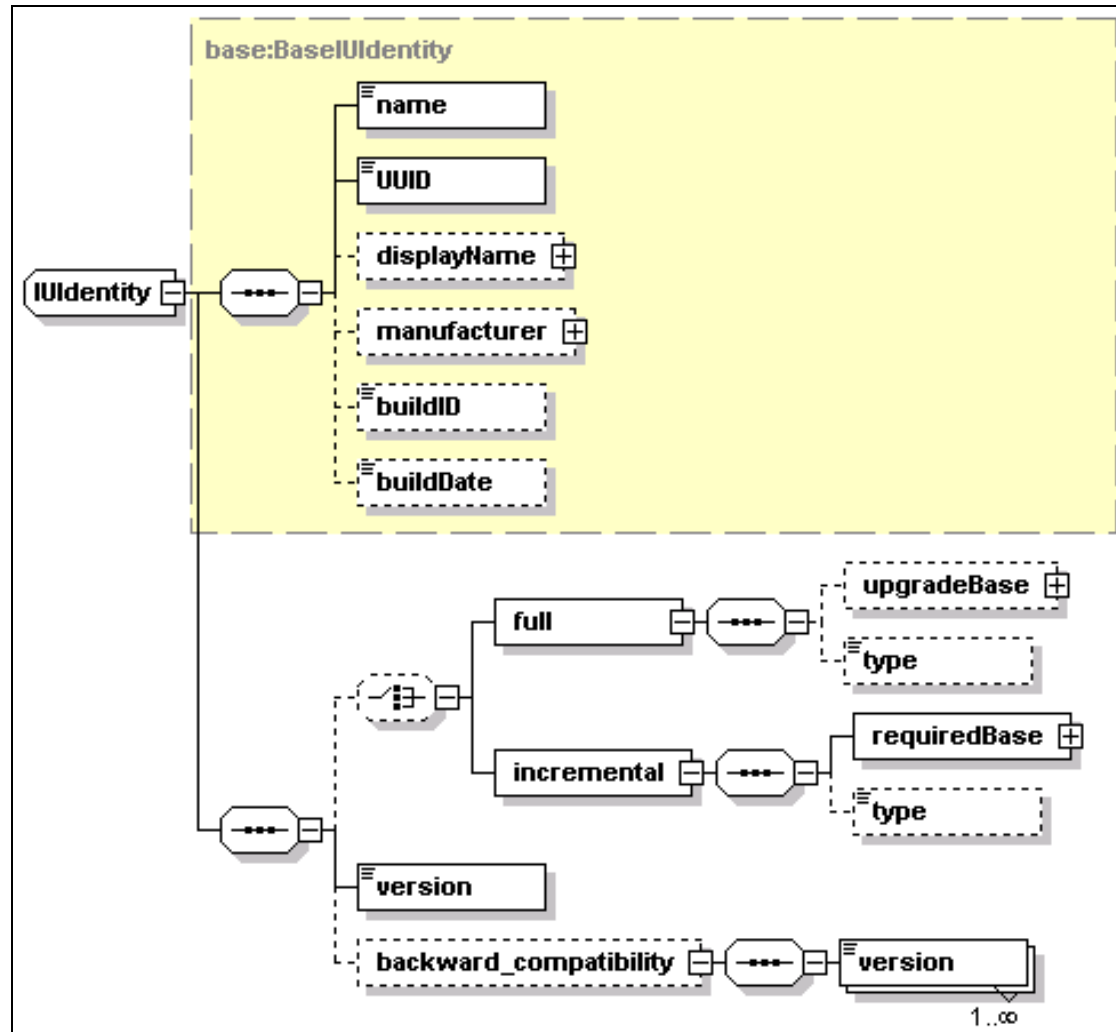
- **SIU (Smallest installable Unit)**
 - Leaf node in the hierarchy
 - Targeted to a hosting environment target (*WindowsOS*)
 - Identity, and dependencies
 - Deployable content (artifacts)
 - An aggregate IU (CIU,SM, root) does not have deployable content of its own.
- **Install Artifact**
 - Defines actions to be executed on the HE (*InstallMSIProduct*)
 - Referenced in the SIU definition as an external file
 - Artifact schema is HE specific
 - Actions in an artifact may reference bundled files (the MSI package in the example)





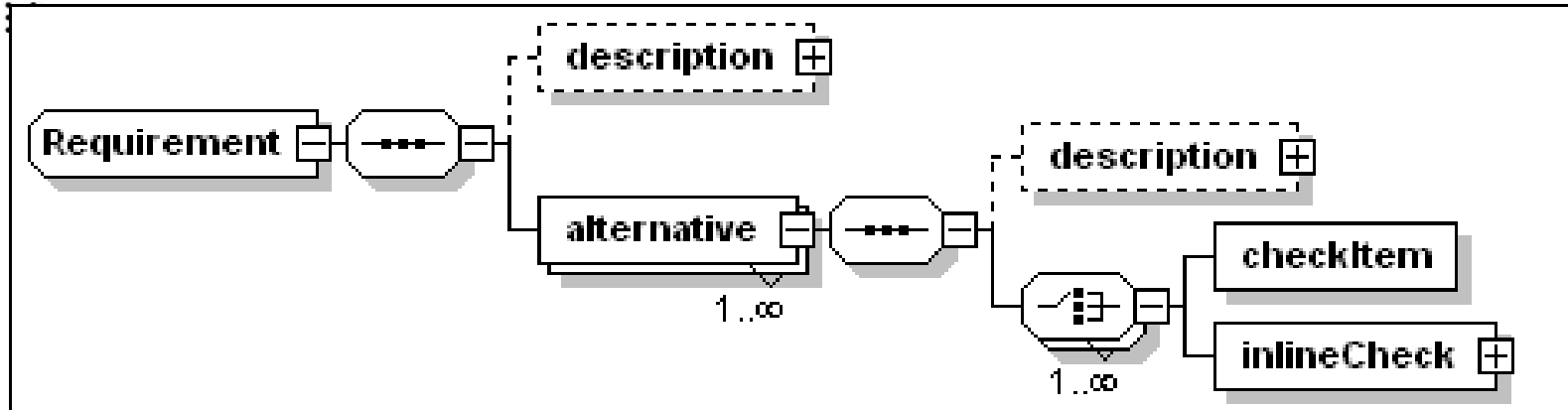
IU Identity

- **IU Identity**
 - name
 - Universally Unique Identifier (UUID)
 - Company/Manufacturer information
 - Build information
- **Base or Update**
 - Full vs incremental update
- **Version**
 - backward_compatibility
- **IU type identity elements**
 - UUID
 - Version
- **Multiple instances allowed**





Requirements (expression of a dependency)



- A requirement is declared to be met by one or more *alternatives* (logical “OR”)
 - At least one alternative must be satisfied
 - Multiple non-exclusive alternatives should indicate a *priority*
 - *Example:* a requirement could be to have either DB2 or MSSQL installed
 - Requirements can be declared in single-target IUs (SIU/CIU) and topology targets
- An alternative combines one or more elementary *checks* (logical “AND”)
 - check definition may be elsewhere (normally in the checks section of the IU) or inline
 - The boolean result of an alternative is the boolean .AND. of all the referenced or inline checks
 - A check can be computed in the alternative with the .NOT. (testValue=“false”)



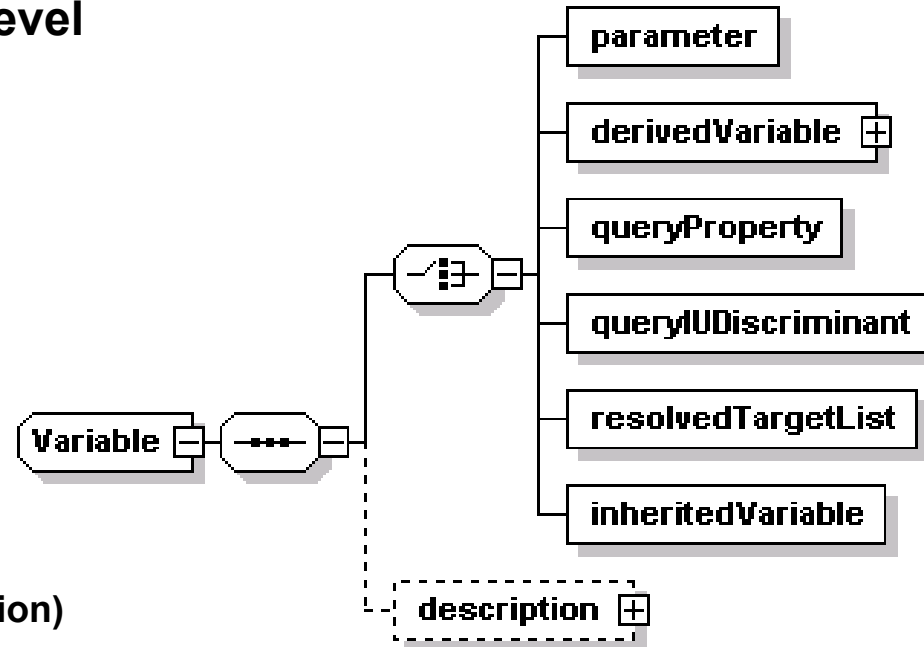
Checks (testing an environment condition)

- **Capacity** of the hosting environment. This specifies a property of the hosting environment, such as processor speed, and some minimum or maximum value.
- **Consumption** of resources allocated on the hosting environment. This specifies resource, such as disk space, that will be consumed by the installation. Consumption requirements are cumulative across installable units.
- **Property** check. This compares the value of a named property exposed by the hosting environment against a specified value or pattern.
- **Version** of a target resource. This is used to check the value of a version property against a specified interval.
- **Software** check. This specifies the check of a software resource, identified by name and version range. The resource may not have an associated IU definition.
- **IU** (Installable Unit) check. This is used to determine if a given IU is installed.
- **Relationship** check. It specifies a relationship that must exist between two topology targets.
- **Custom** checks. These specify the execution of a user defined command.



IU Variables

- Variables can be defined at any IU level
- Variable types
 - *parameter*
 - May have default
 - Overridable (e.g. via a response file)
 - *derivedVariable*
 - Conditioned variable expressions
 - *queryProperty*
 - Value from topology target
 - *queryIUDiscriminant*
 - IU instance identifier (i.e. install-location)
 - *resolvedTarget list*
 - List of target instances for a logical target
 - *inheritedVariable*
 - To get the persisted value of a variable from an existing instance being updated





Variable expressions and conditions

- **Variable Expression**
 - Symbolic reference to one or more variables
 - Example: “\$(rootDirectory)/MyProductDirectory”

- **Boolean variable expressions are used to condition the following**
 - **Installable units within the IUDD**
 - If “false”, the IU is ignored
(selection based on environmental condition vs user selection)

 - **Sets of artifacts associated to a SIU**
 - E.g. to select the artifact set suitable for a given target instance

 - **Initialization of a derived variable (multiple conditioned expressions)**



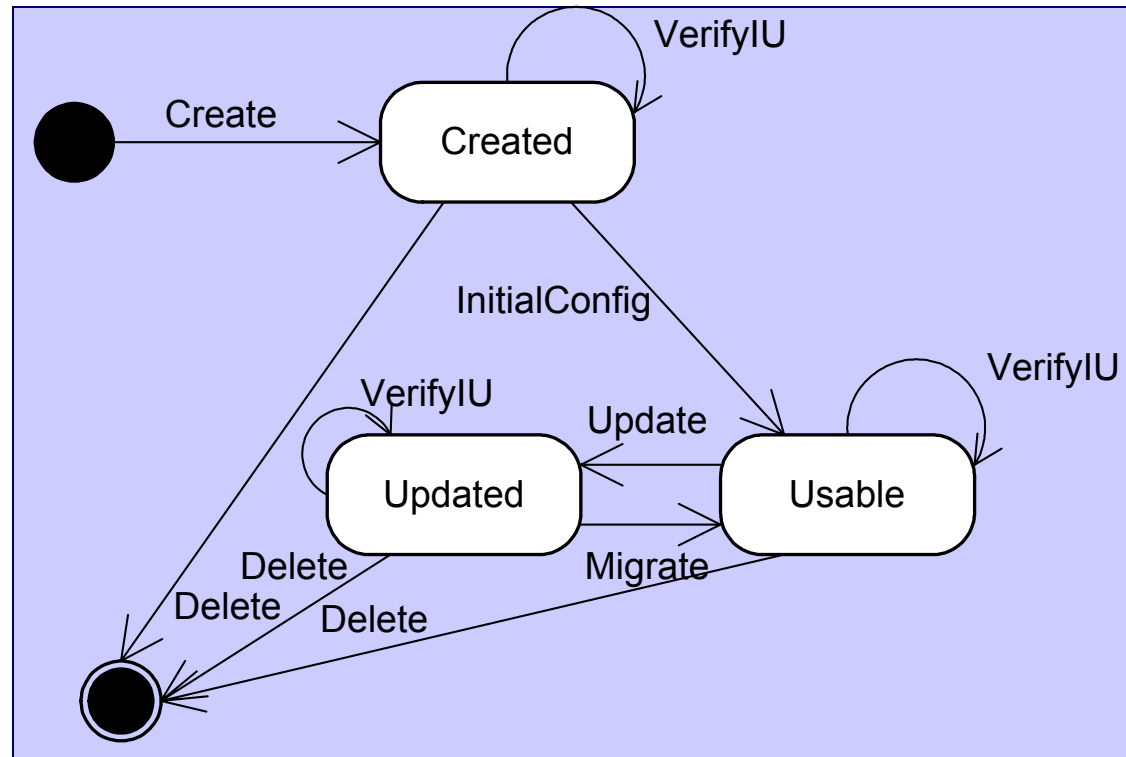
IU life cycle – CM operations and artifacts

Change Management Operations

- Create
- InitialConfig
- Update
- Migrate
- Delete
- VerifyIU
- Undo

IU Artifacts

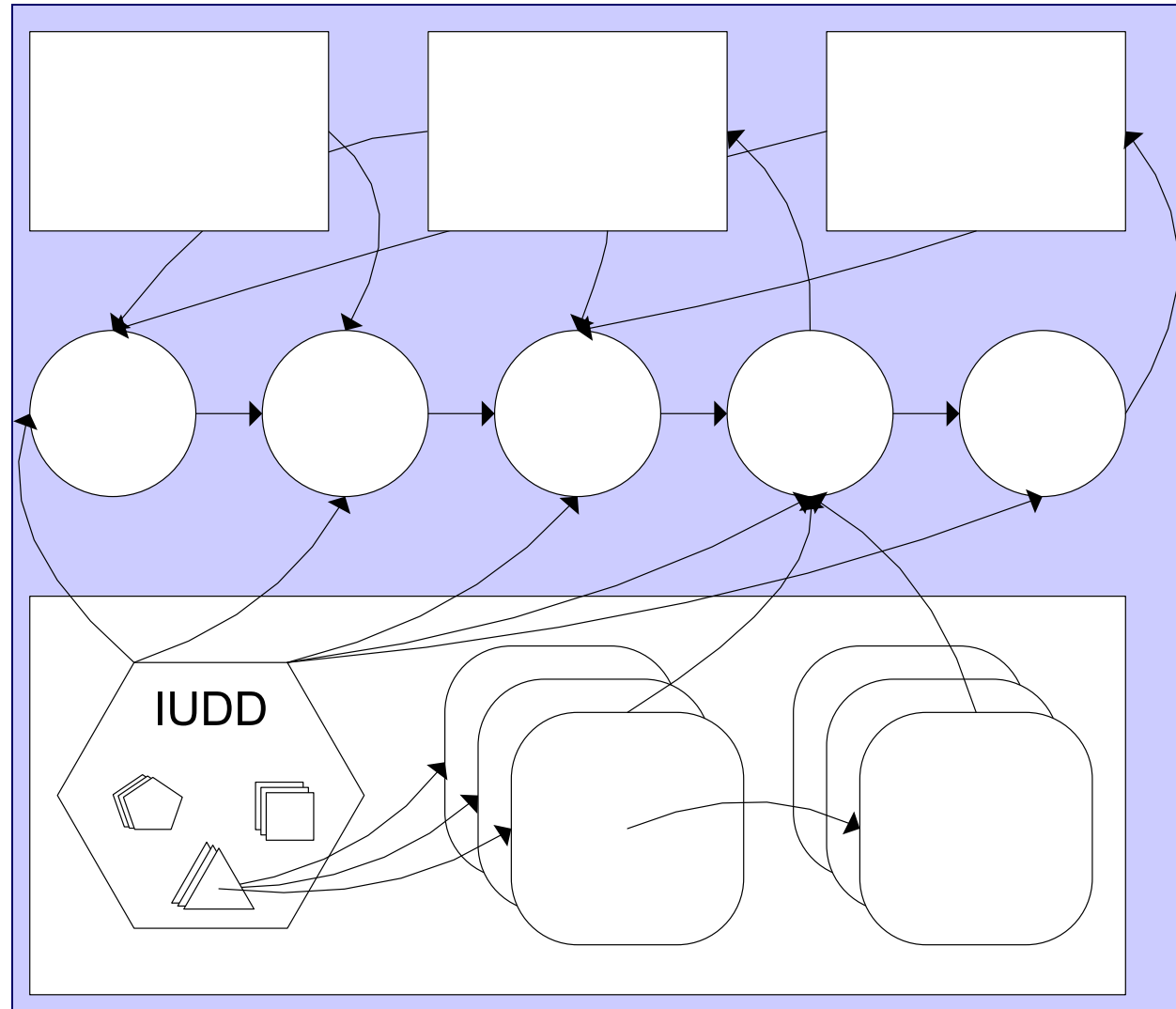
- Most operations have an associated artifact
 - Install artifact [for Create & Update]
 - InitConfig/Migrate artifacts
 - Delete artifact





Logical view of the change management process

- One or more hosting environments are the targets of the software being installed.
- The user provides input, in this process, either interactively or via a response file.
- That input drives the activity of a software installer program that interacts with the hosting environments and the IU registry.





**Installable Unit
Deployment Descriptor
- ACS-WG -
(acs-wg@ggf.org)**



Summary

- **Requirements for IUDD and AAF are very similar**
- **Leverage a broader standard like IUDD so configuration problem is addressed at many levels of granularity with same data**
- **Candidate standards such IUDD and provisioning implementation efforts such as NaReGI and others provide foundation.**
- **Encourage vendors of install products to participate in ACS-WG.**