Updates – what should a recipient do?

• Check for Authenticity?
• Install a payload (or more)?
  • How should it obtain the payload?
  • How should it install?
  • How should it verify?
  • How should it assess applicability?
Descriptive updates

• Provide a recipient information
  • Authenticity
  • Integrity
  • Applicability
  • Representation
  • Encapsulation
Descriptive updates – tradeoffs

• Pros:
  • Data can be compact
  • Data is easy to present to a user

• Cons:
  • Specification of behaviour is weak
    • Behaviour is implied by combinations of data
    • Flags must be added to direct behaviour when not sufficiently implied
    • Many corner cases exist
  • Data becomes complex
    • Structure of data follows organization of features
    • Data is not regular, which adds to recipient complexity.
  • Capability reporting requires separate definition
Behavioural updates

• Instruct a recipient how to obtain, apply, verify, load, and run an update

• All the same information is present, but the organization is restructured into order of consumption, rather than by association
Behavioural updates – tradeoffs

• Pros:
  • Unambiguous
  • Simple data encoding
  • Simple parsing
  • Behaviour is explicit, not implied
  • High flexibility

• Cons
  • Size may be larger (not necessarily)
  • More tooling required
Behavioural manifest outer structure

- Authentication Wrapper
- Manifest
  - Externally accessible data
  - Common data
  - Command-lists
Behavioural manifest command lists

• 6 command lists, 1 for each stage in lifecycle:
  • Dependency-resolution
  • Image acquisition
  • Image application
  • System verification
  • Image loading
  • Image invocation

• 1 Common command list
  • Processed before each other command list
Behavioural manifest dependency handling

• Command lists are processed in lock-step between dependencies
  • For example, image acquisition for all manifests is processed before any image installation is processed.

• Dependencies are explicitly processed.
Behavioural manifest commands

Conditions
• Check device identity
• Verify image presence (correctness) or absence
• Verify dependency availability
• Check component
• Check system
• Check 3rd-party authorisation

Directives
• Process sub-behaviours
• Process dependencies
• Set parameters
• Move an Image or Document
• Invoke an Image
• Wait for an event
Imperative manifest parameters

- Strict Order
- Soft-Failure
- Source List (defines move source)
- Processing Step Configuration (affects move)
- Image Identifier