A Binary Manifest Serialization Format for IoT Devices

draft-pagel-suit-manifest-00

Martin Pagel

7 Nov 2018

Agenda

- Design Philosophy
  - Simple to process and small
- Update Server
- Manifest Format
- Implementation

Simple to Process

- Binary format
  - no parser necessary
- Provide framework, details are platform specific, like:
  - Component types and payload formats
  - Memory location, preprocessing, installation
  - Encryption, digest, signature format, key mgmt

Small Memory Requirements

- No need for download buffer
- Fewer options and instructions
- Only Image URIs
  - separate image download

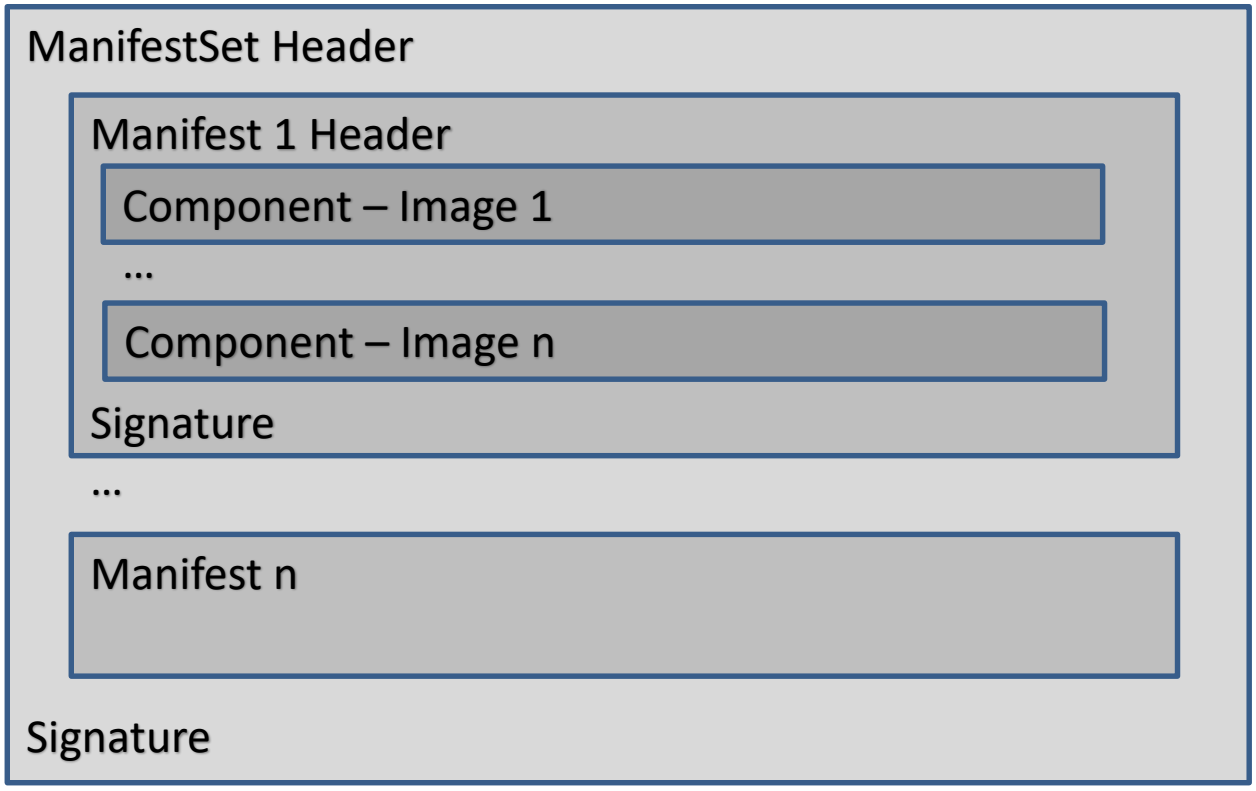
While Still Powerful

- Multiple images
  - Aggregate different origins
  - Each signed separately
- ABI versions and dependency graph

Update Server

- Facilitates Network/Device/OEM Operator
- Policy Enforcement
  - Targeting, Rollout speed, Rollbacks, Locations
  - Authorization and Dependency checks
- Platform specific Manifest Handling
- Image Download via CDN (or broadcast)
  - Device only downloads new images
  - Short-lived URL for Privacy

Manifest Format Diagram



Implementation

- ImageType for:
  - Payload type
  - Signature method
  - Installation order
  - Storage location
  - Preprocessing
- BuildDate instead of sequence number

Questions

ManifestSetHeader

Type	Field	Description
Uint32	MagicValue	0x7086760e acting as a static file format signature
Uint16	Version	1 - Version of the manifest set data structure
Uint16	Flags	Hints for device specific policy engine, it can either be interpreted as 16 flags, integer value, or a combination depending on the device
Uint16	ManifestSetDataSize	Size of the total set in bytes

ManifestFooter

Type	Field	Description
Uint8[20]	SignCertThumbprint	Thumbprint of the cert used to sign this manifest. All zeros if the manifest is unsigned.
Uint8[64]	Signature	Digital signature of all the data prior to this field using the signature method specific to the device/platform.

Manifest

Type	Field	Description
UInt16	Version	Version of the manifest data structur e
UInt16	ImageCount	Number of images in the manifest
UInt16	ManifestEntrySize	Size of each entry in bytes, allows safe interpretation even if size changes due to data structure version changes
UInt8[16]	VendorId	UUID5(DNS, "example.com")
UInt8[16]	ClassId	UUID5(VendorId, "Product X")
UInt64	BuildDate	Manifest creation time in unix epoch time
ImageManifestEnt ry[ImageCount]	ImageEntries	Entries for the images
UInt8[20]	SignCertThumbprint	Thumbprint of the cert used to sign this manifest. All zeros if the manifest is unsigned.
UInt8[64]	Signature	Digital signature of all the data prior to this field using the signature method specific to the device/platform.

ImageManifestEntry

Type	Field	Description
UInt8[16]	ImageUid	Image UID
UInt8[16]	ComponentUid	UID of the Component the image represents.
UInt16	Type	Component Type (values specific to the device architecture)
UInt32	CompressedImageFileSize	Size of the image file in bytes as compressed
UInt32	UncompressedImageFileSize	Size of the image file in bytes after it is uncompressed
ABIDependency[2]	Provides	Lists any ABI type and version this component provides
ABIDependency[2]	DependsOn	Lists any ABI type and version this component it consumes meaning depends on

ABIDependency

Type	Field	Description
UInt32	Version	Image UID
UInt32	ABIType	Type of ABI interface