A Canonical Event Log Structure for IMA

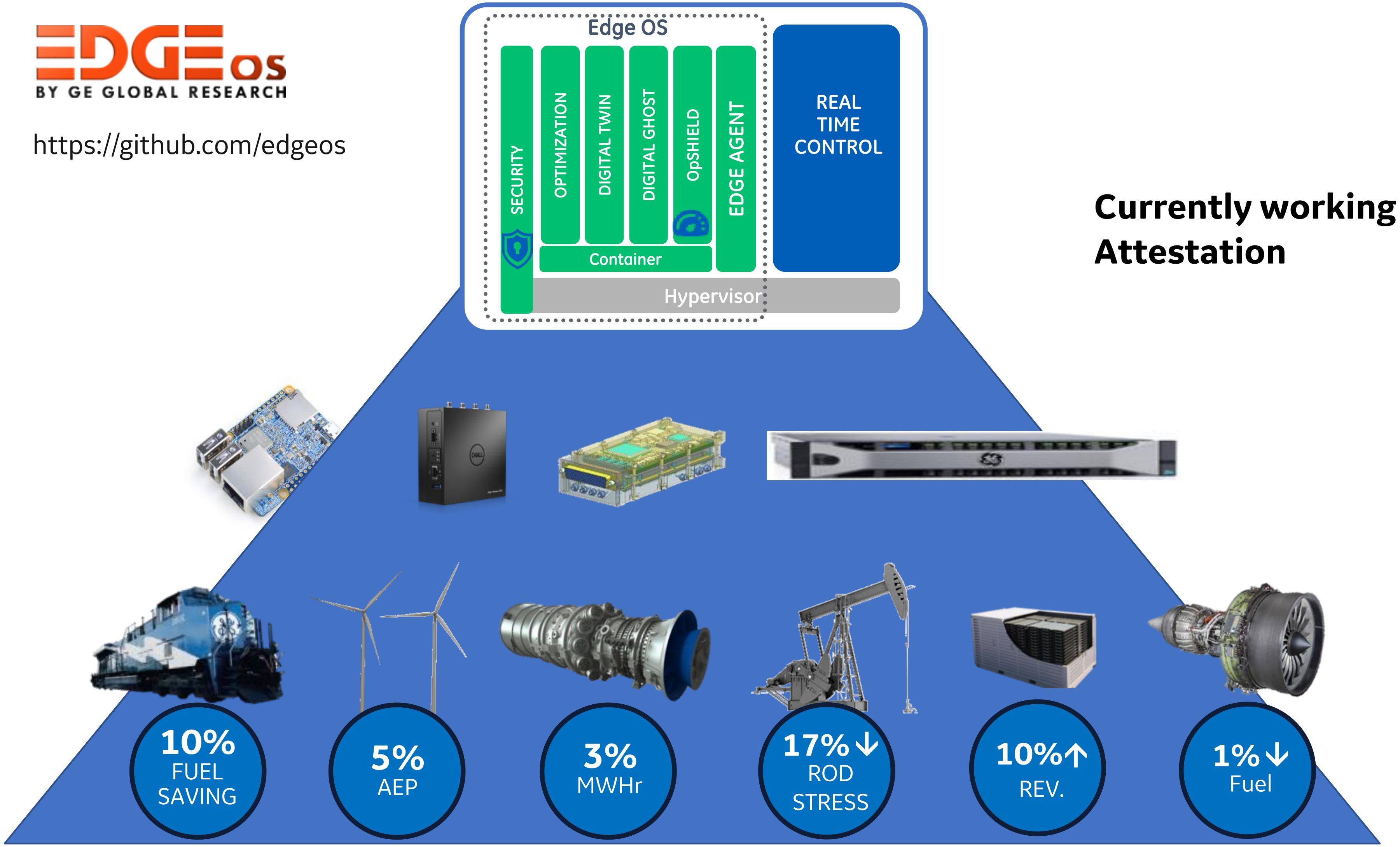
David Safford and Monty Wiseman
General Electric

August 28, 2018

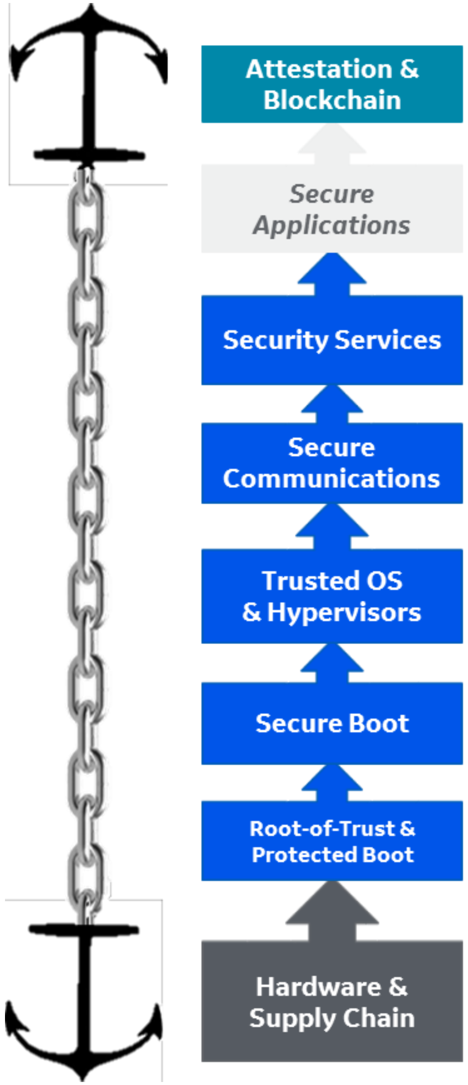
Secure, Software Defined Analytics & Controls Platform

EDGE OS
BY GE GLOBAL RESEARCH

https://github.com/edgeos



Currently working on Attestation



IMA Measurement List Desires

1. Scalability!
    - The existing measurement list (and hash table) are big kernel memory leaks
    - The leaks are made worse by incessant violation records (millions - Lawrence Reinert/NCSC)
    - **There is no reason for measurements to remain in the kernel – they're TPM protected**

2. Completeness!
    - We need to attest metadata, since metadata can affect security and IMA policy
    - Owner, group, mode, security labels
    - Make it _much_ easier to add fields in the future

3. Standards compliance!
    - Canonical Event Log Format for IMA and other logs
    - A common TLV format will make integrated attestation simpler and more robust
    - If we are going to make a new list format, we might as well directly support the TCG format

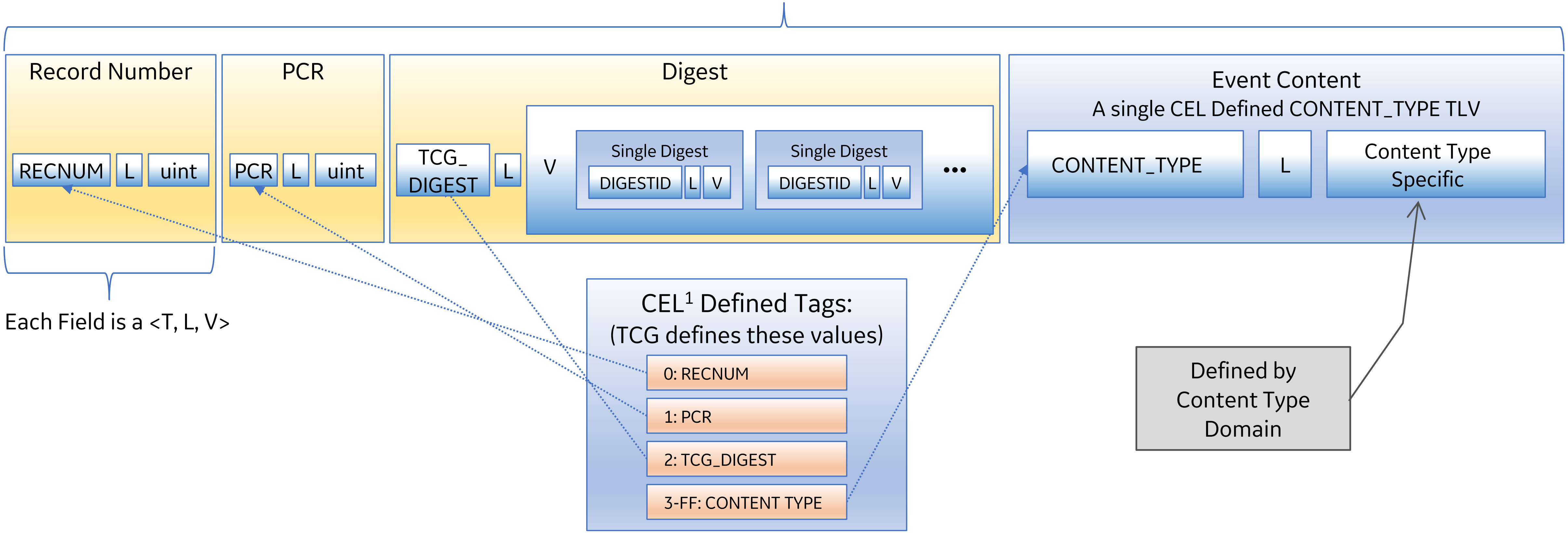
4. Bonus!
    - Sequence number for attestation compression/synchronization
    - Timestamp for better event correlation
    - Flexible/Dynamic selection of included fields
    - Don't need to transfer measurement list across kexec!

	Local Appraisal	Remote Attestation
Data	**IMA**	**IMA**
Metadata	**EVM**	**???**

Canonical Event Log Record

One Canonical Event Log Record

Field Cohesion required within each Event Log Record



Record Number

RECNUM | L | uint

PCR

PCR | L | uint

Digest

TCG_ DIGEST | L | V

Single Digest: DIGESTID | L | V

Single Digest: DIGESTID | L | V

•••

Event Content

A single CEL Defined CONTENT_TYPE TLV

CONTENT_TYPE | L | Content Type Specific

Each Field is a <T, L, V>

CEL[1] Defined Tags:

(TCG defines these values)

0: RECNUM

1: PCR

2: TCG_DIGEST

3-FF: CONTENT TYPE

Defined by Content Type Domain

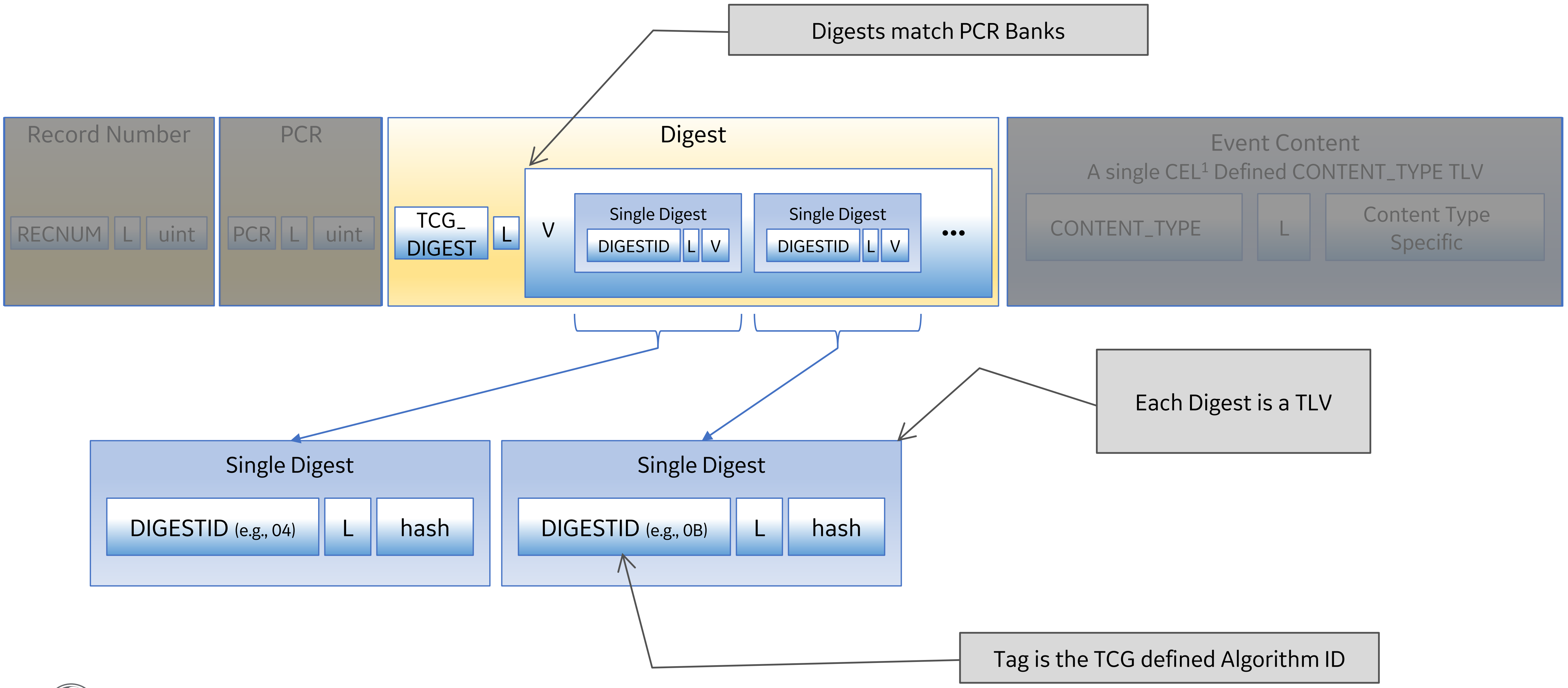
[1] CEL: Defined by the TCG Canonical Event Log specification

CEL Tags and Tag Ranges

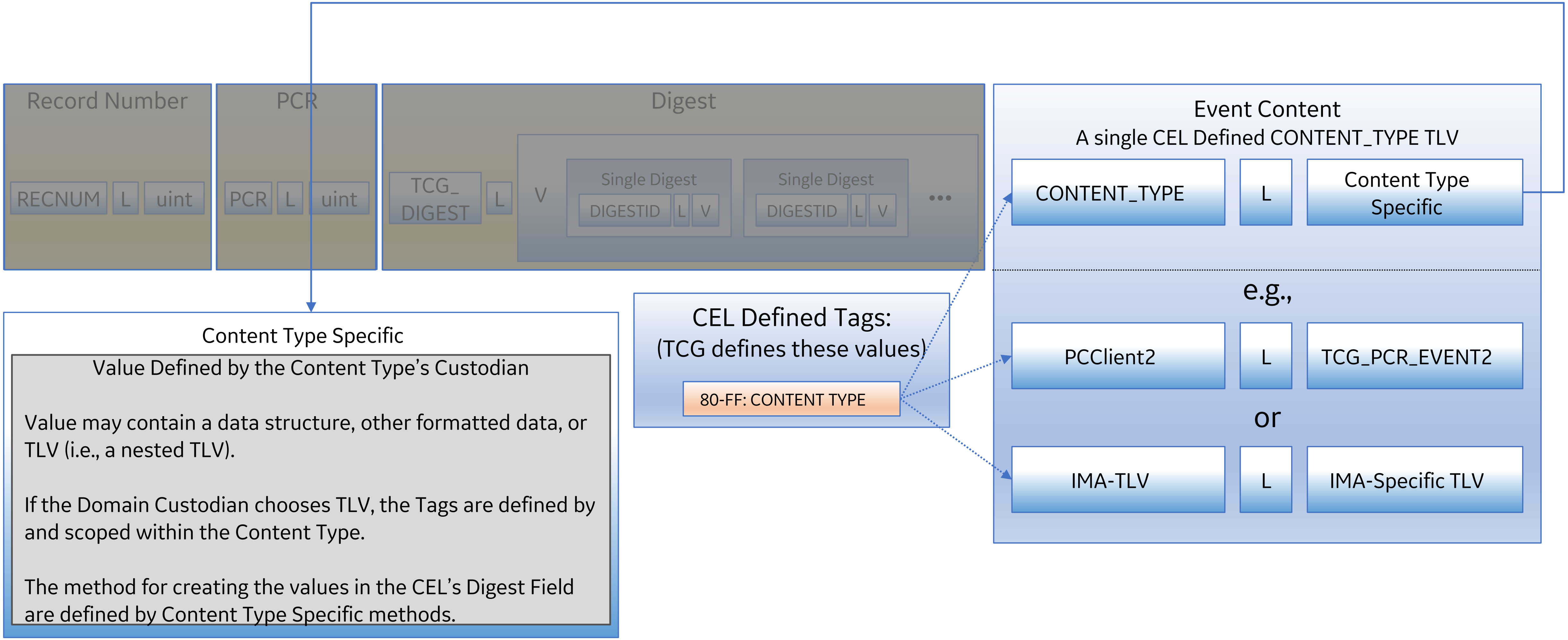
Constant	TCG Tag	Description of Value Field
00[1]	RECNUM	Unique Record Number
01	PCR	PCR Index[2]
02	TCG_DIGEST	TCG Digest
03-	CONTENT_TYPE	Type of Content

Constant	CONTENT Type	Description
3	CEL	Content managed by TCG / CEL. Provides information & management of log
4	PCClient-STD	PC Client WG defined encapsulating structure
5	PCClient-TLV	PC Client WG defined using TLV (?)
6	IMA-Legacy	IMA Legacy
7	IMA-TLV	IMA TLV

Digest Field



Event Content Field



Record Number

RECNUM | L | uint

PCR

PCR | L | uint

Digest

TCG_ DIGEST | L | V

Single Digest: DIGESTID | L | V

Single Digest: DIGESTID | L | V

…

Event Content

A single CEL Defined CONTENT_TYPE TLV

CONTENT_TYPE | L | Content Type Specific

e.g.,

PCClient2 | L | TCG_PCR_EVENT2

or

IMA-TLV | L | IMA-Specific TLV

CEL Defined Tags:

(TCG defines these values)

80-FF: CONTENT TYPE

Content Type Specific

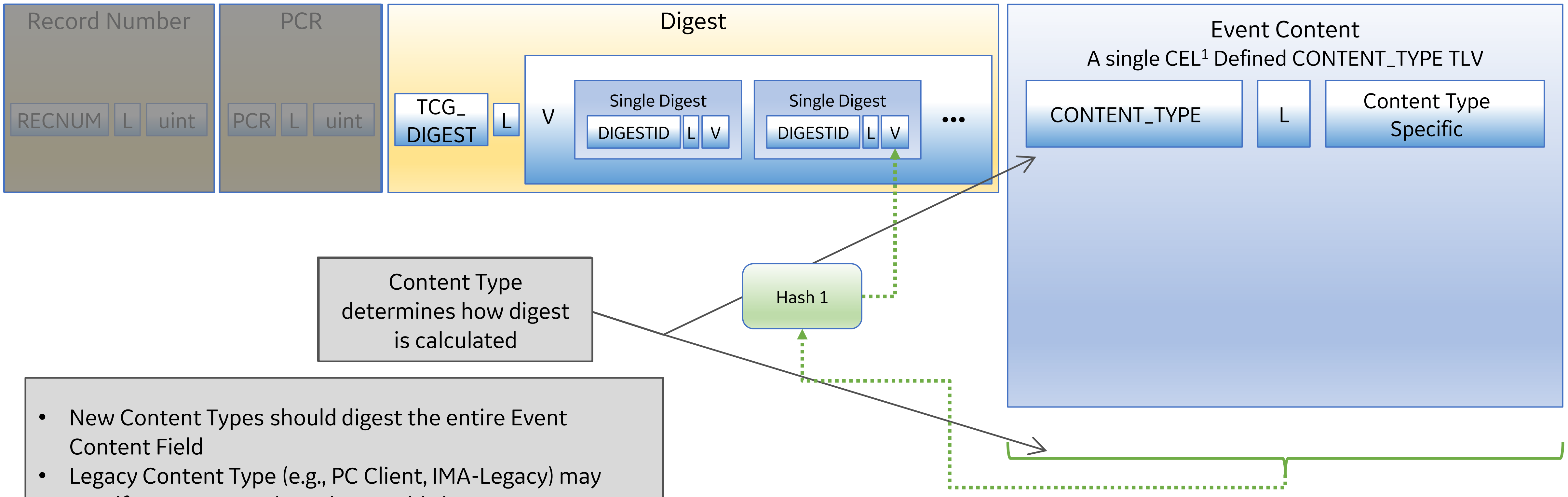
Value Defined by the Content Type’s Custodian

Value may contain a data structure, other formatted data, or TLV (i.e., a nested TLV).

If the Domain Custodian chooses TLV, the Tags are defined by and scoped within the Content Type.

The method for creating the values in the CEL’s Digest Field are defined by Content Type Specific methods.

Digest Field Calculation



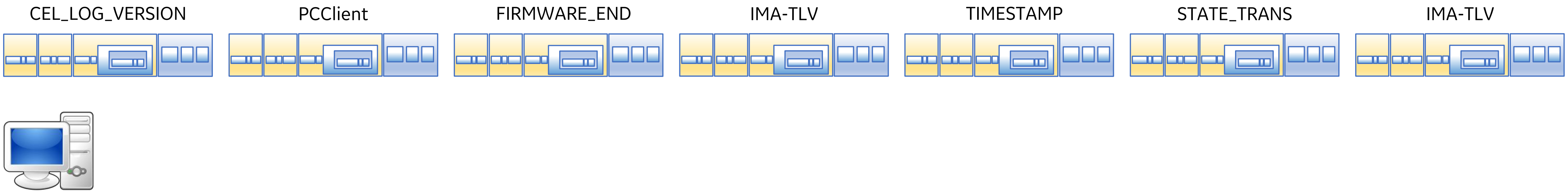
Content Type determines how digest is calculated

- New Content Types should digest the entire Event Content Field
- Legacy Content Type (e.g., PC Client, IMA-Legacy) may specify a more complex scheme. This is necessary to maintain consistent digest to content mapping.

CEL Management

- Used to Provide information about the Event Log
  Version information -- which spec version
  Time stamp
  Separators (e.g., between firmware and OS)
- Similar to PC Client EV_Separator
- This content is managed / maintained by TCG
- Security-sensitive events are measured
  E.g., Time stamps
- Non-security-sensitive events are not measured
  E.g., Event log (spec) version info

CEL Event Log Example Sequence



CEL_LOG_VERSION

PCClient

FIRMWARE_END

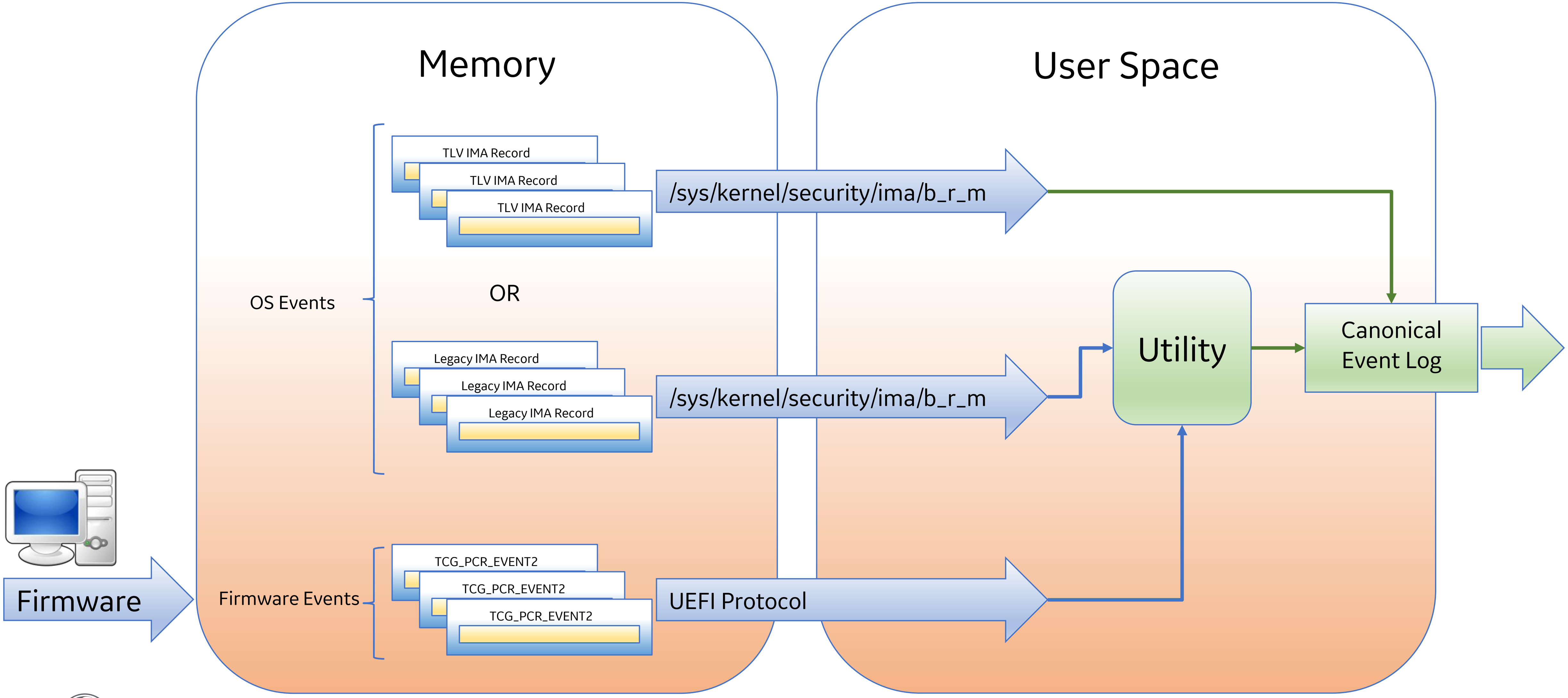
IMA-TLV

TIMESTAMP

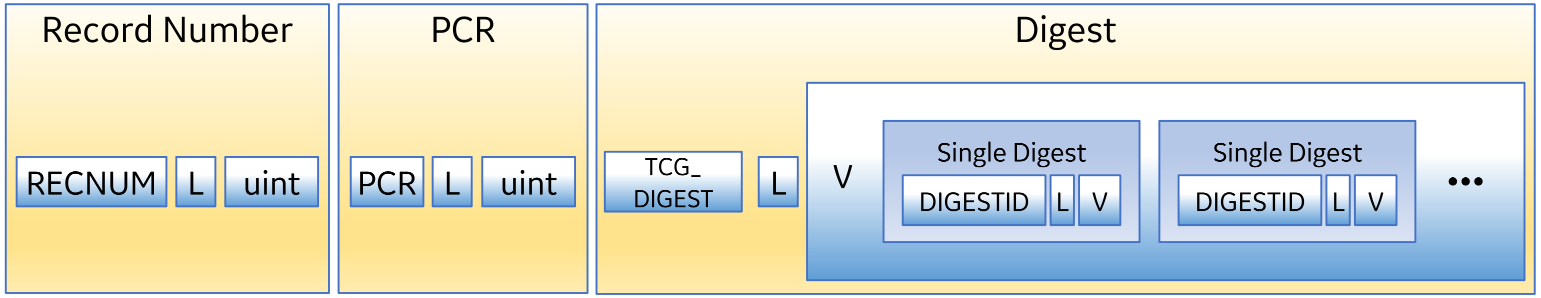
STATE_TRANS

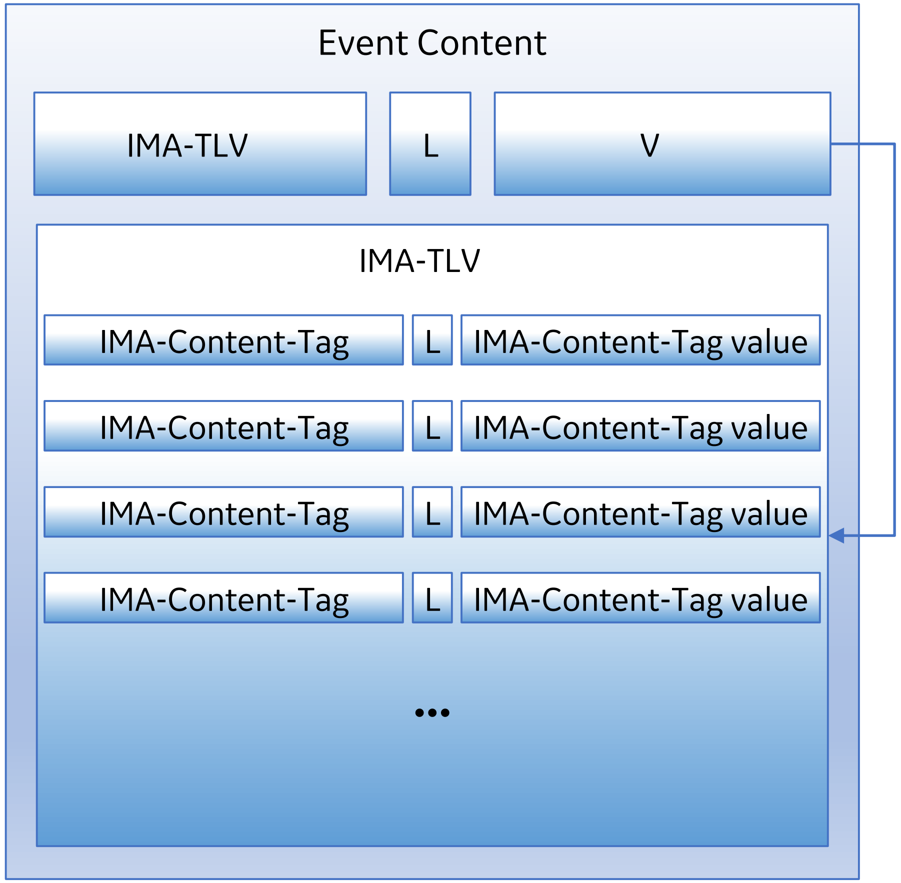
IMA-TLV

Example Event Log Management



IMA-TLV



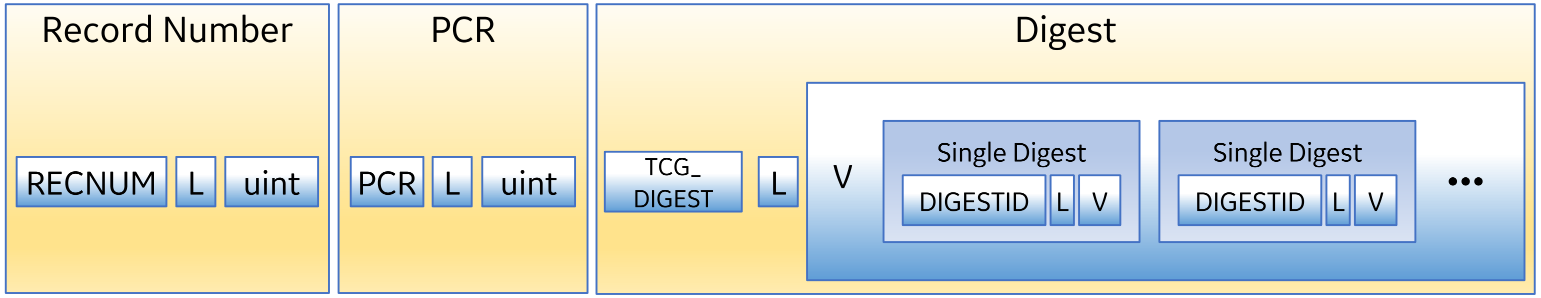


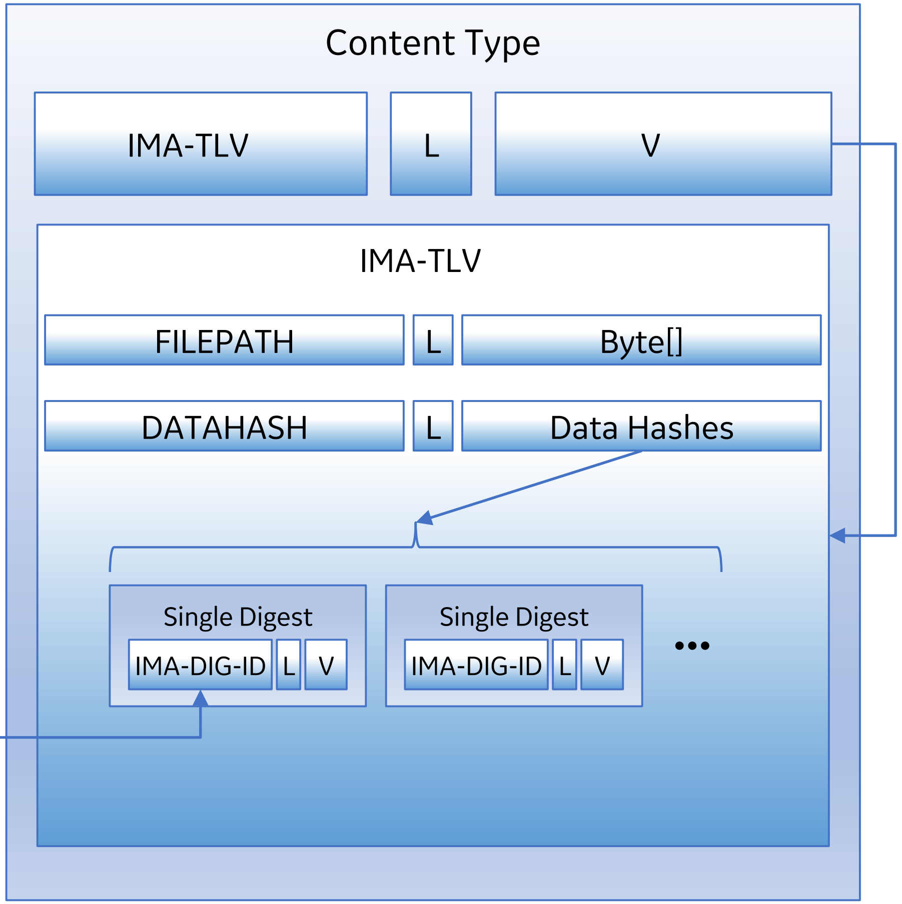
Constant	IMA-Content-Tag	Comment
0	FILEPATH	Primitive
1	DATA	Primitive
2	DATAHASH	V is array of Single Digest TLVs
3	DATASIG	V is array of Single Signature TLVs
4	METADATAHASH	V is array of Single Digest TLVs
5	METADATASIG	V is array of Single Signature TLVs
6	OWNER	Primitive
7	GROUP	Primitive
8	MODE	Primitive
9	LABEL	Primitive
A	TIME_STAMP	Primitive
B...	...	...

Defined by IMA (i.e., Constant values may overlap CEL defined constants)

Recommend using same format as CEL Mgmt

IMA-TLV Example Event Log



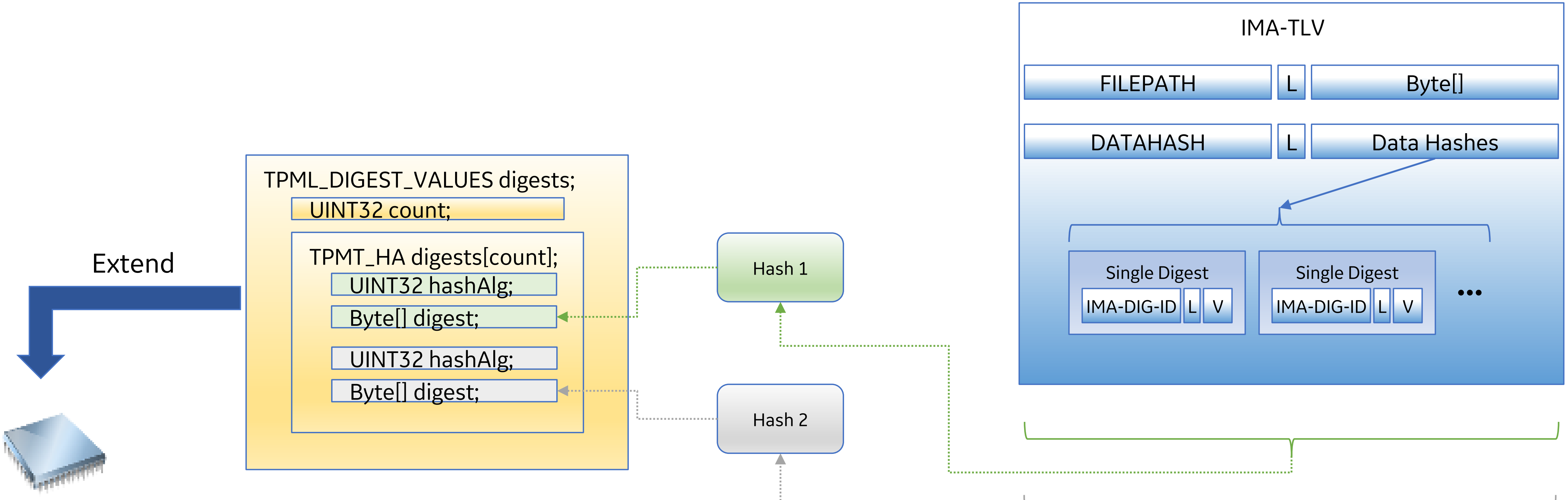


Constants are examples only. These are not the TCG-defined Constants. These are aligned with IMA Digest IDs

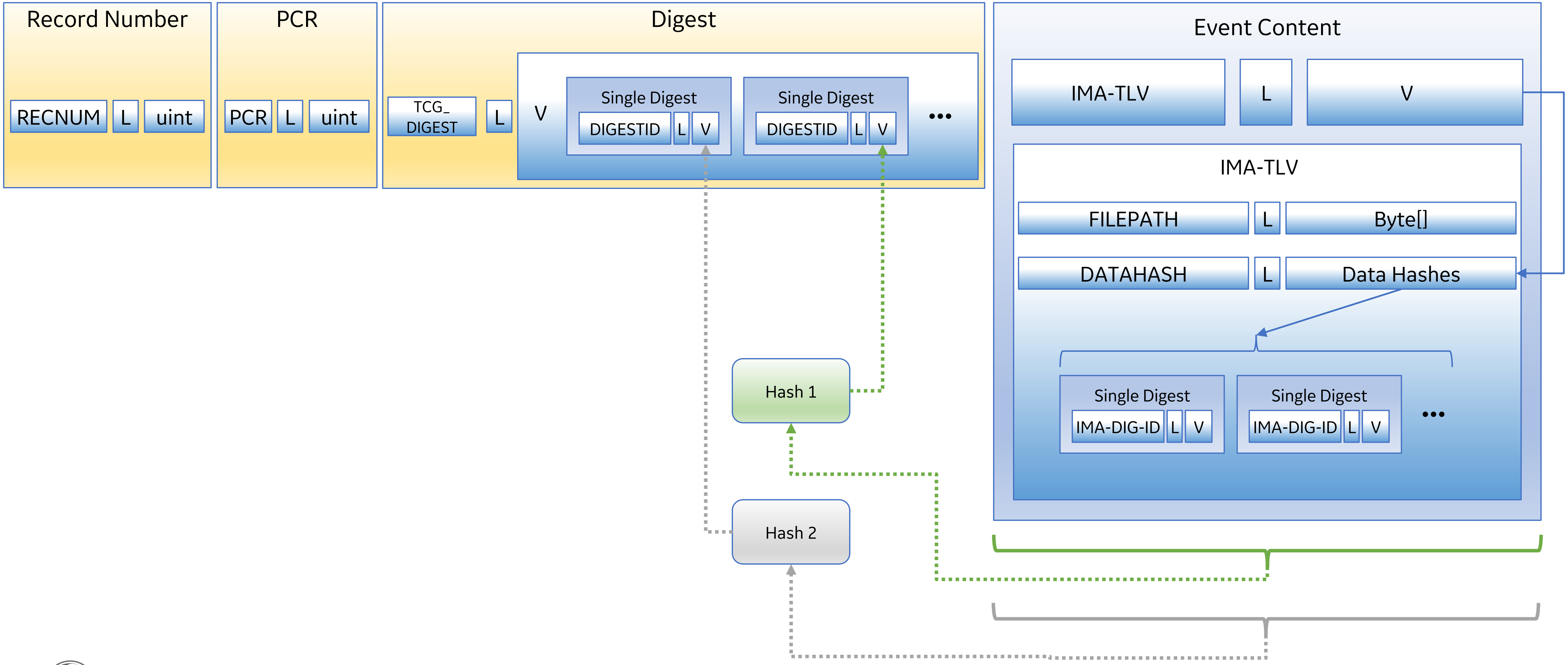
IMA Defined Digest ID

Constant	Algorithm Name
10	TPM_ALG_SHA or TPM_ALG_SHA1
11	TPM_ALG_SHA256
12	TPM_ALG_SHA384
13	TPM_ALG_SHA512
14	TPM_ALG_SM3_256
15	TPM_ALG_SHA3_256
16	TPM_ALG_SHA3_384
17	TPM_ALG_SHA3_512
18 – 7F	Reserved

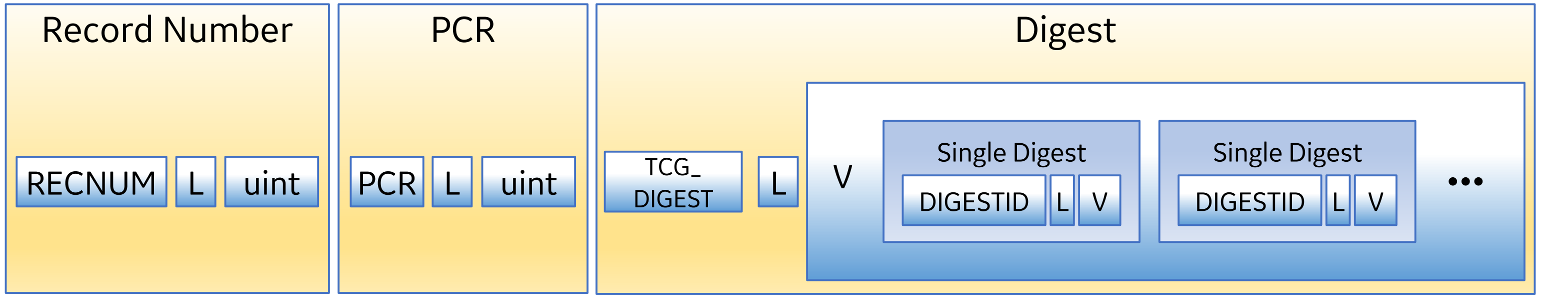
IMA-TLV Measurement



IMA-TLV Record Hash



IMA-Legacy



Record Number

RECNUM | L | uint

PCR

PCR | L | uint

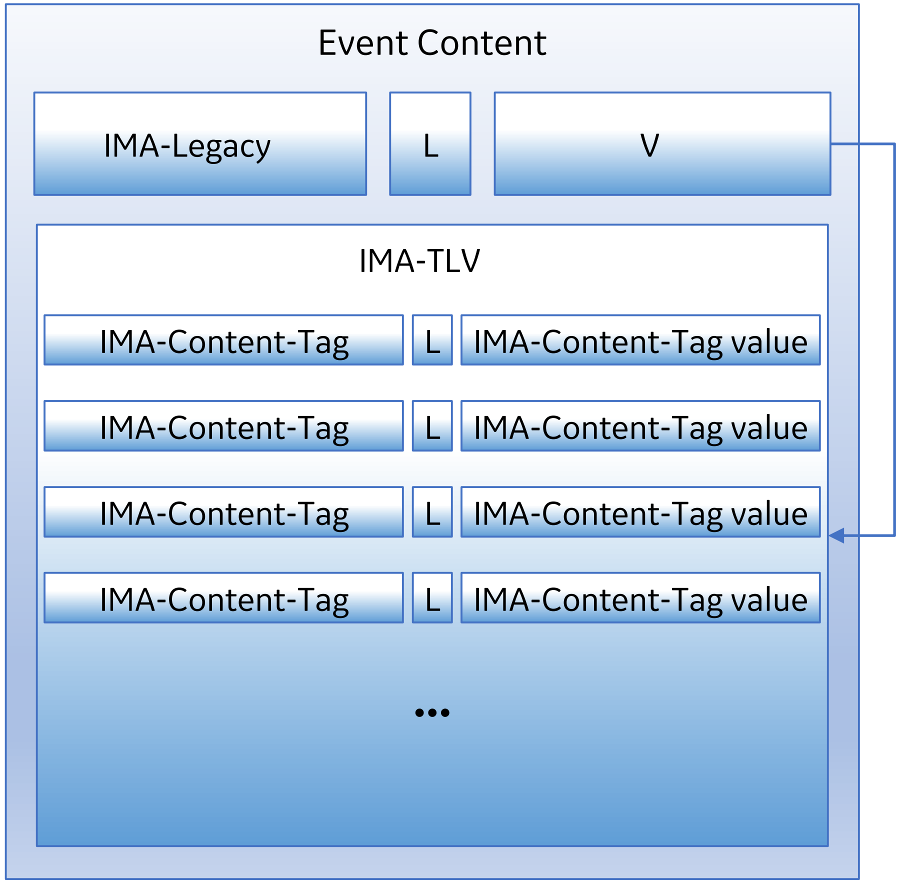
Digest

TCG_ DIGEST | L | V

Single Digest: DIGESTID | L | V

Single Digest: DIGESTID | L | V

•••



Event Content

IMA-Legacy | L | V

IMA-TLV

IMA-Content-Tag | L | IMA-Content-Tag value

IMA-Content-Tag | L | IMA-Content-Tag value

IMA-Content-Tag | L | IMA-Content-Tag value

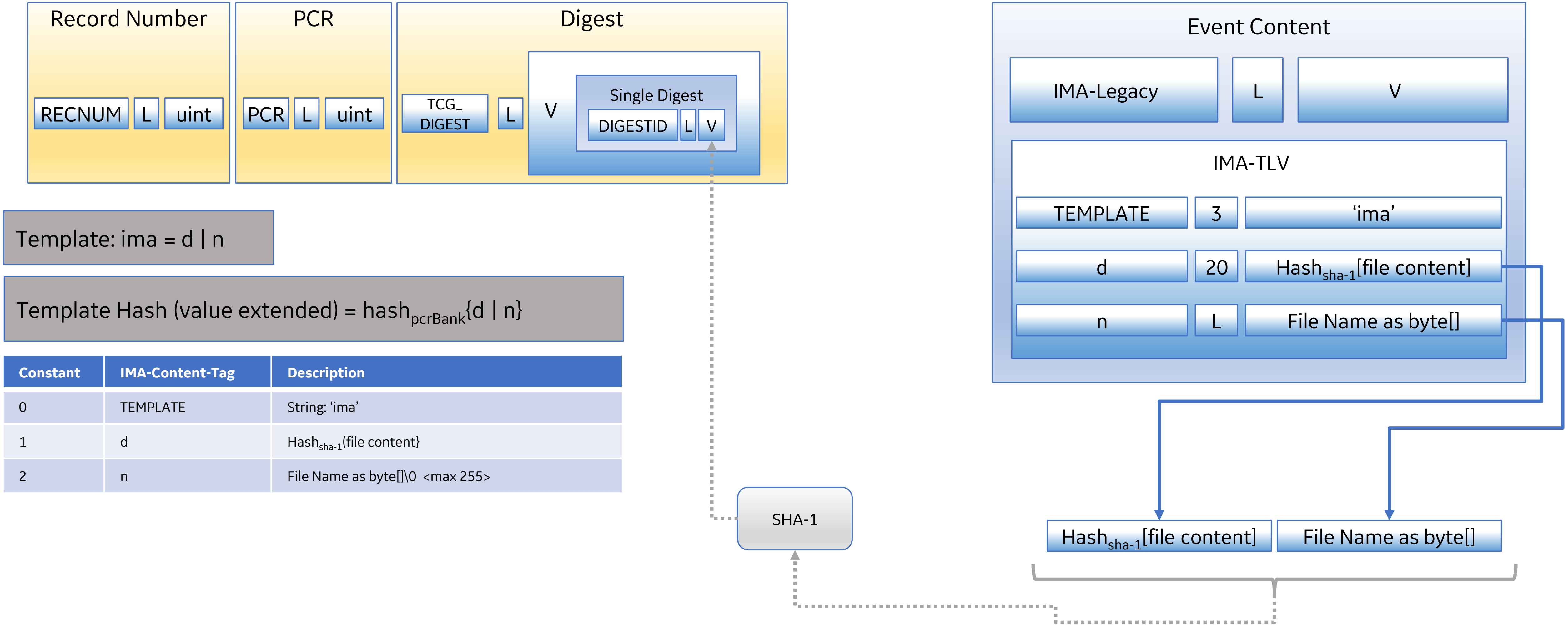
IMA-Content-Tag | L | IMA-Content-Tag value

•••

Constant	IMA-Content-Tag	Description
0	TEMPLATE	String: ‘ima’ | ‘ima-ng’ | ‘img-sig’
1	d	Hash$_{\text{sha-1}}$(file content}
2	n	File Name as byte[]\0 <max 255>
3	d-ng	“SHA-1:” | hash$_{\text{sha-1}}$[file content] “SHA-256:” | hash$_{\text{sha-256}}$[file content] …
4	n-ng	UINT16 Len | file name as byte []
5	sig	file signature as byte[]

Defined by IMA
(i.e., Constant values
may overlap CEL
defined constants)

IMA-Legacy – Template: ima



Template: ima = d | n

Template Hash (value extended) = $\text{hash}_{\text{pcrBank}}$\{d | n\}

Constant	IMA-Content-Tag	Description
0	TEMPLATE	String: 'ima'
1	d	$\text{Hash}_{\text{sha-1}}$\{file content\}
2	n	File Name as byte[]\0 <max 255>

PoC/RFC Patchset

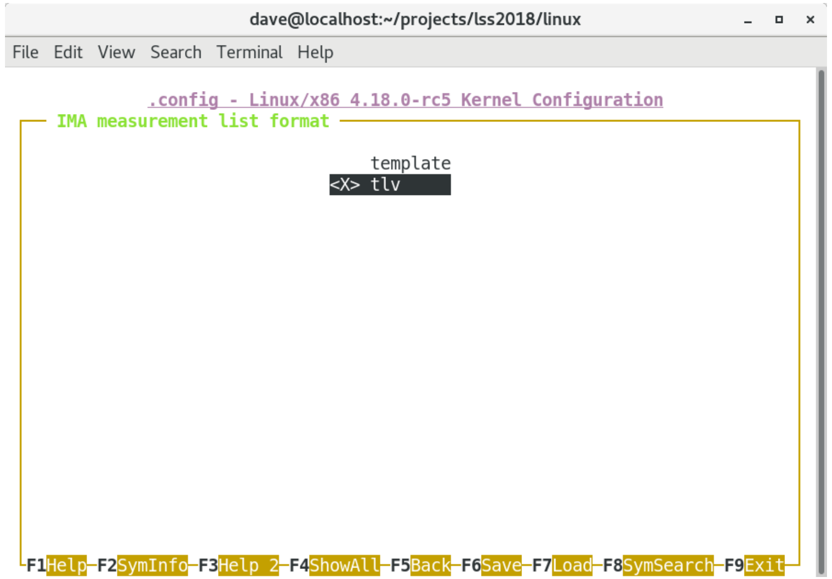
1. Refactoring – new abstraction (“records”) pointing to either TLV or Template
2. Currently config/compile time selection of format, TLV-or-Template
3. Measurement list is truncated on read, hash table is omitted.

Common:
280 ima_api.c
466 ima_appraise.c
634 ima_crypto.c
322 ima_fs.c
289 ima.h
141 ima_init.c
494 ima_main.c
55 ima_mok.c
1209 ima_policy.c
3890

6134 total

Template:
230 ima_fs_template.c
201 ima_queue_template.c
640 ima_template.c
88 ima_template.h
390 ima_template_lib.c
45 ima_template_lib.h
170 ima_kexec.c
1764

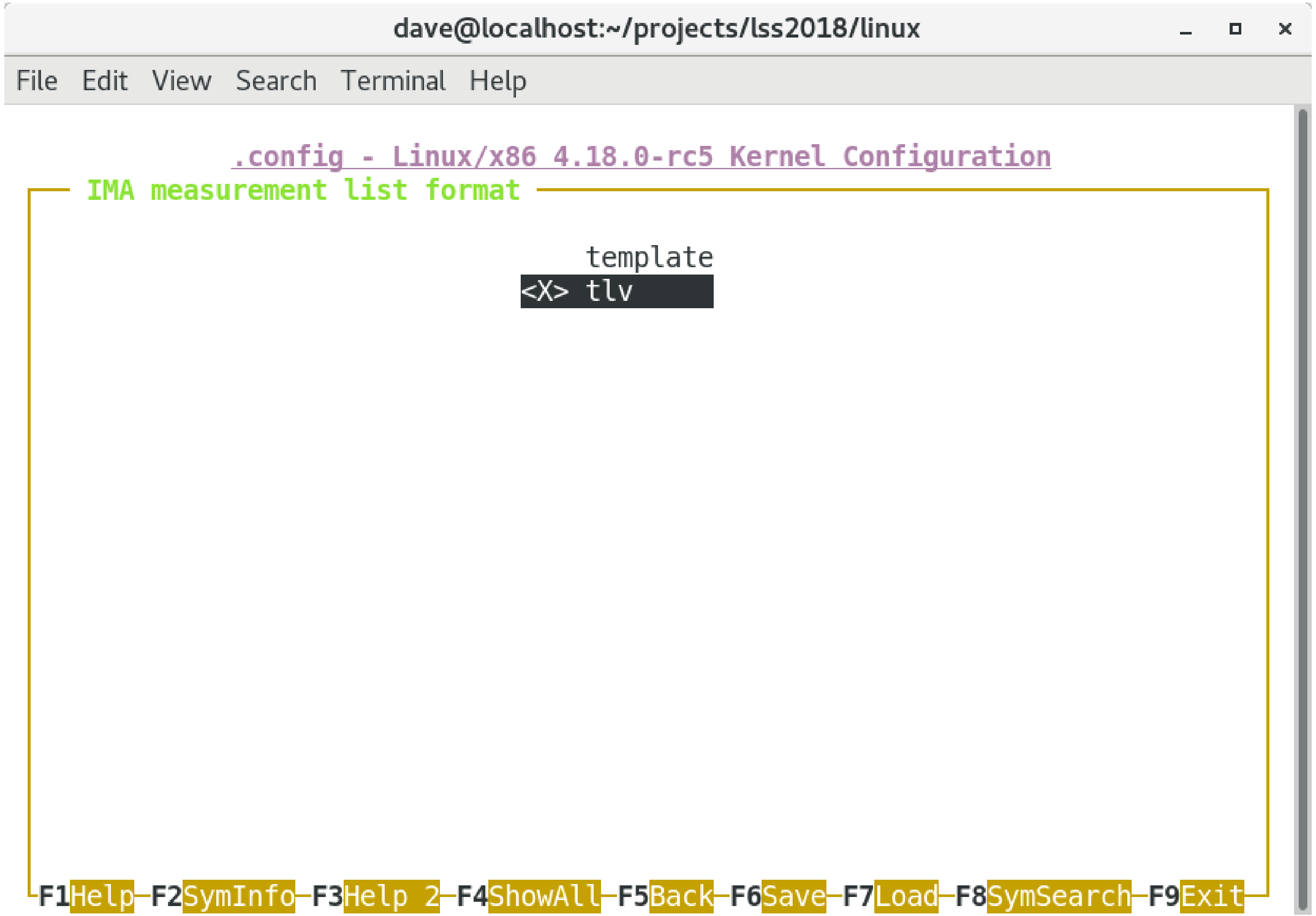
TLV
105 ima_fs_tlv.c
95 ima_queue_tlv.c
205 ima_tlv.c
75 ima_tlv.h
480


dave@localhost:~/projects/lss2018/linux
File Edit View Search Terminal Help
F1Help F2SymInfo F3Help 2 F4ShowAll F5Back F6Save F7Load F8SymSearch F9Exit

```
.config - Linux/x86 4.18.0-rc5 Kernel Configuration
 IMA measurement list format

                    template
                <X> tlv
```

PoC/RFC Patchset - kconfig



dave@localhost:~/projects/lss2018/linux
File Edit View Search Terminal Help

```
           .config - Linux/x86 4.18.0-rc5 Kernel Configuration
┌─ IMA measurement list format ──────────────────────────────────────┐
│                                                                    │
│                             template                               │
│                         <X> tlv                                    │
│                                                                    │
└F1Help─F2SymInfo─F3Help 2─F4ShowAll─F5Back─F6Save─F7Load─F8SymSearch─F9Exit┘
```

Adding a field

```
/* IMA Specific Content Types */
#define IMA_TLV_CONTENT_PATH         0
#define IMA_TLV_CONTENT_DATAHASH     1
#define IMA_TLV_CONTENT_DATASIG      2
#define IMA_TLV_CONTENT_OWNER        3
#define IMA_TLV_CONTENT_GROUP        4
#define IMA_TLV_CONTENT_MODE         5
#define IMA_TLV_CONTENT_TIMESTAMP    6
#define IMA_TLV_CONTENT_LABEL        7

code to calculate length of the new TLV:
        if (is_selected(IMA_TLV_CONTENT_MODE) && inode)
                l = l + IMA_TLV_HDR_SIZE + sizeof(inode->i_mode);

code to fill in TLV data:
        if (is_selected(IMA_TLV_CONTENT_MODE) && inode) {
                ima_tlv_buf(pos, IMA_TLV_CONTENT_MODE, sizeof(inode->i_mode),
                           (const u8 *)&(inode->i_mode));
                pos = pos + IMA_TLV_HDR_SIZE + sizeof(inode->i_mode);
        }
```

Ima_tlv_selected=<bitmask>

3 ~= ima-ng

7 ~= ima-sig

Demo

```
cat /sys/kernel/security/ima/tlv_runtime_measurements > bindata
cat bindata | ./tlv_dump
...
SEQNUM 00001364 PCRNUM 10 TCG_DIGEST SHA1
280634A43216E1BB7438130B6A6ED95F9ED6F909 PATH
/home/dave/projects/lss2018/tlv_dump DATA_HASH SHA256
D6F820A121A111D2952FBCE85B2985634344A5B57C521A2A267375BE672AE4B6
Digest Matches content <==
Final pcr-10 should be 0E5FB2405486563DC059D43A3D28BD9AF21647DE <==
[root@localhost dave]# tpm2_pcrlist
sha1 :
  0  : c38713029d7433a7be8c5a89dc8660bef5e37899
  1  : 056ad82d0d2e20f3c6541ed67debd0e534c63f55
  2  : b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236
  3  : b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236
  4  : 220bc46deadfb67faea90c92bece61d6400bbf87
  5  : 647586e80172debe28540589cb252e8cf4ef5570
  6  : b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236
  7  : ca5525bfdb2f10814dbd4f69364698b6d8b35dd9
  8  : fde0e8dd62dddf802403091966004b21e4398182
  9  : 04337e9370fb1bc022ecc0e1ef60a18845efad35
  10 : 0e5fb2405486563dc059d43a3d28bd9af21647de <==
```

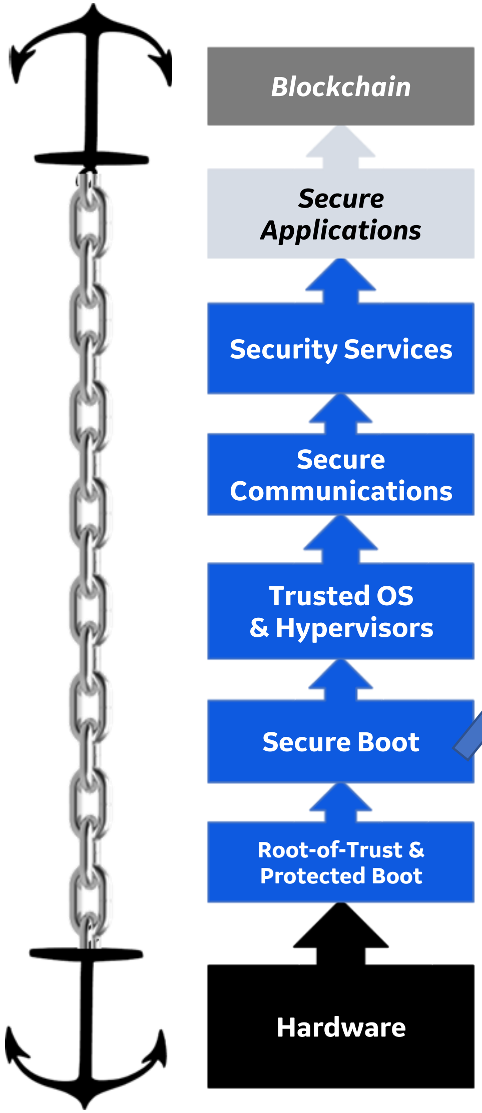
Summary - RFC

1. Desires
    a) Eliminate Memory Leak
    b) Attest metadata
    c) Simplify writing and parsing measurement list
    d) Standards Compliance

2. RFC:
    a) Any reason for TLV-and-Template?
    b) Any in-kernel need for hash table?
    c) Long Term - Deprecate Template?
    d) Other fields desired?
    e) Other comments/suggestions?

Extra Credit

NanoPi Neo Plus2 - A Secure *Pi Platform ($35)



Blockchain

Secure Applications

Security Services

Secure Communications

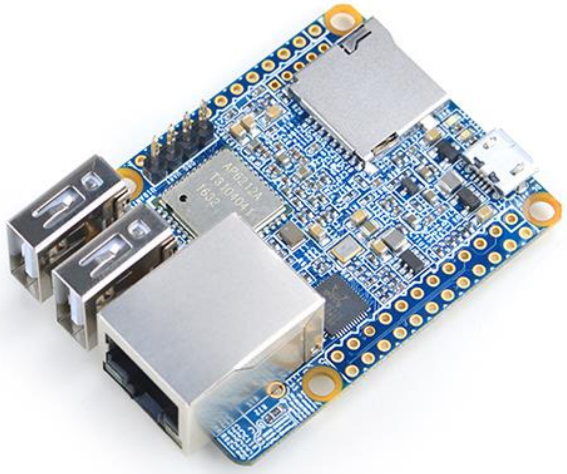
Trusted OS & Hypervisors

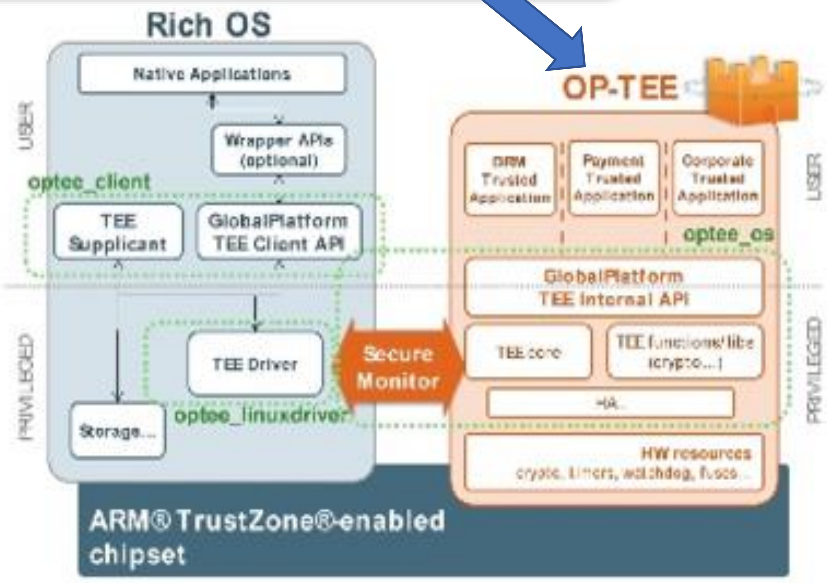
Secure Boot

Root-of-Trust & Protected Boot

Hardware

	Raspberry Pi	BeagleBone Black	NanoPi
Protected boot	none	emmc wp	emmc wp
Verified Boot	none	U-boot	U-boot
TPM	none	none	Trustzone fTPM




Rich OS
Native Applications
Wrapper APIs (optional)
optee_client
TEE Supplicant
GlobalPlatform TEE Client API
USER
PRIVILEGED
TEE Driver
Storage...
optee_linuxdriver
Secure Monitor
OP-TEE
DRM Trusted Application
Payment Trusted Application
Corporate Trusted Application
optee_os
GlobalPlatform TEE Internal API
TEE core
TEE functional libs (crypto...)
HAL
HW resources crypto, timers, watchdog, fuses...
USER
PRIVILEGED
ARM® TrustZone®-enabled chipset