

arm

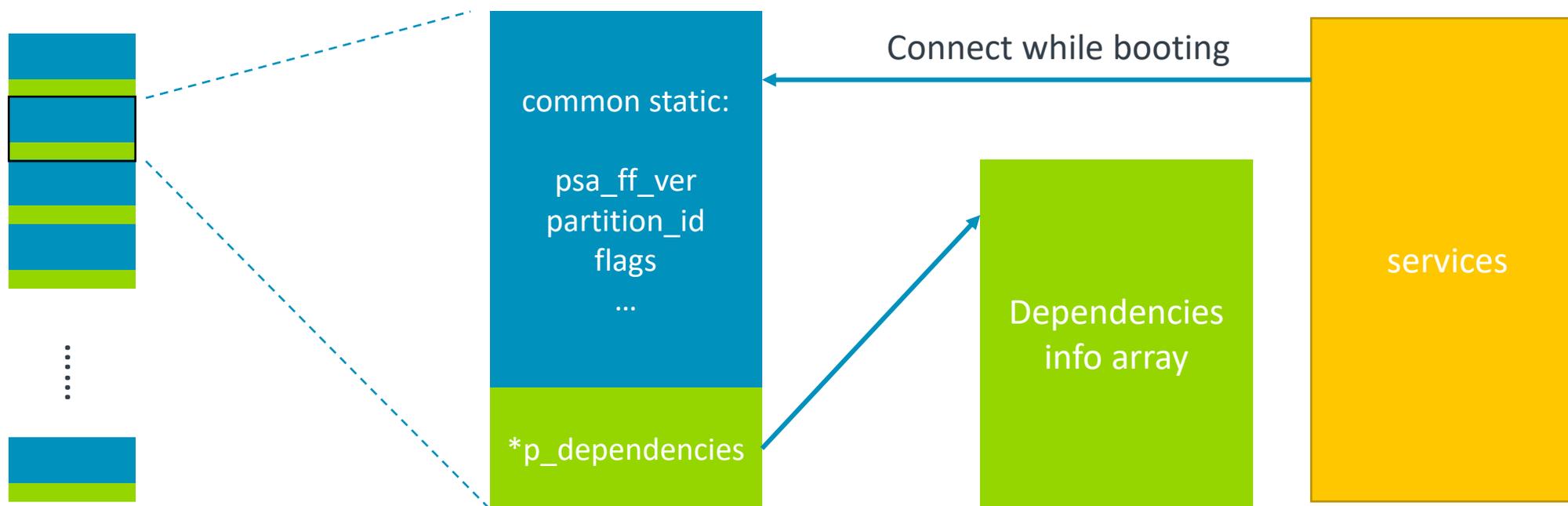
Partition Storage Arrangement



Mingyang Sun
2020-11-12

Partition storage arrangement

- Current situation
 - An overall SP database contains all the partitions' information.
 - Use pointers to link data structures – such as `*p_dependencies` to connect dependencies.
 - Partition info and service info in separate files, use runtime enumeration to connect them while booting.

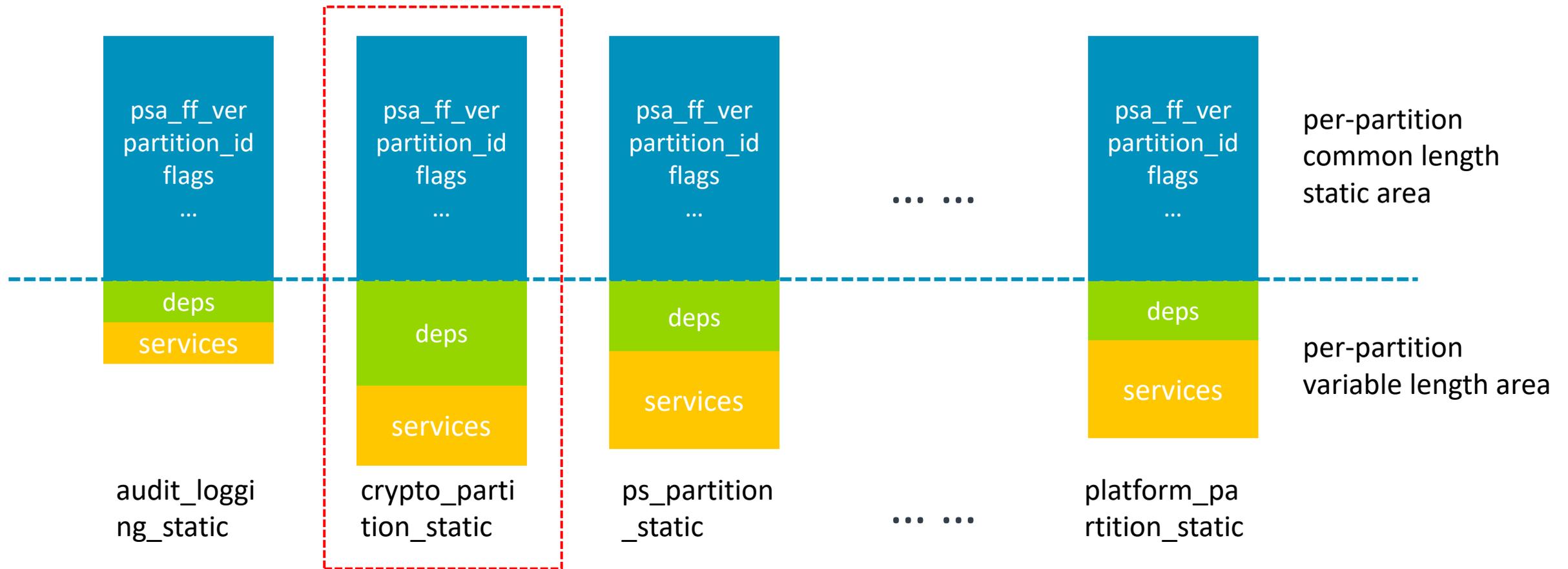


Partition storage arrangement

- Restrictions
 - Pointer like *p_dependencies won't work if relocation happens.
 - Service data in a separate file "tfm_service_list.inc", not easy for reviewing.

- Goal
 - A more straightforward and easier way to organize the partition, deps and service information.
 - Simplified runtime initialization.
 - Easier way to transform into other shapes for future extension. (Eg., Standalone partition library or binary).

Partition structure type for each partition



- Create a structure type for each partition
- save service, dependencies with partition basic static info together, to show the relationship them.

Partition structure type for each partition

crypto_partition_info.h:

```
struct crypto_partition_storage_t {  
    /* common static data */  
    .....  
    uint32_t ndeps = NUM_DEPS;  
    uint32_t size;  
    /* per-partition variable length area */  
    uint32_t deps[NUM_DEPS];  
    struct service_static_t services[NUM_SERVICES];  
}  
struct service_static_t { /* service data members */ };
```

These values are private inside partition sources.



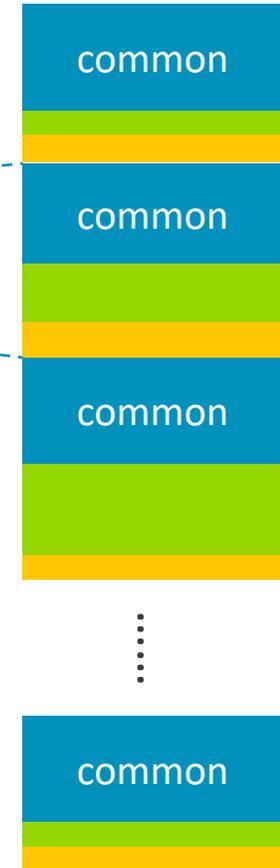
- Different numbers of deps and services for different partitions — SPM does not need to know the actual table. The variable members are connected with runtime data by loader.

Partition data storage – the first iteration

Put all partition storage structures into one struct.

```
struct partition_list_t {  
    struct audit_logging_storage_t audit_logging_static;  
    struct crypto_partition_storage_t crypto_partition_static;  
    struct ps_partition_storage_t ps_partition_static;  
    .....  
    /* other partition type and storage object */  
}
```

- Easily extended into other shapes: a plain storage on storage device.
- Accessed by enumeration.



To sum up

- SP storage
 - A simpler way to organize partition, service, deps data.
 - Easy to use and extend.
 - Manage variable length data in structure.
- Future expansion
 - Each partition's struct type can be regarded as 'partition head' and put as the head of standalone partition binary.
 - Apply to other scenarios.

arm

Thank You

Danke

Merci

谢谢

ありがとう

Gracias

Kiitos

감사합니다

धन्यवाद

شكرًا

ধন্যবাদ

תודה