MSSA

Multi-Service Storage Architecture

- **Three file abstractions:**

  flat, structured, continuous medium

- **A file is built from byte segments**

  different byte segment implementations
  may be used to cater for
  different file characteristics

- **Files are named by storage-service identifiers SSIDs**

  -used as embedded references
  -applications can choose their
  own(textual)naming schemes

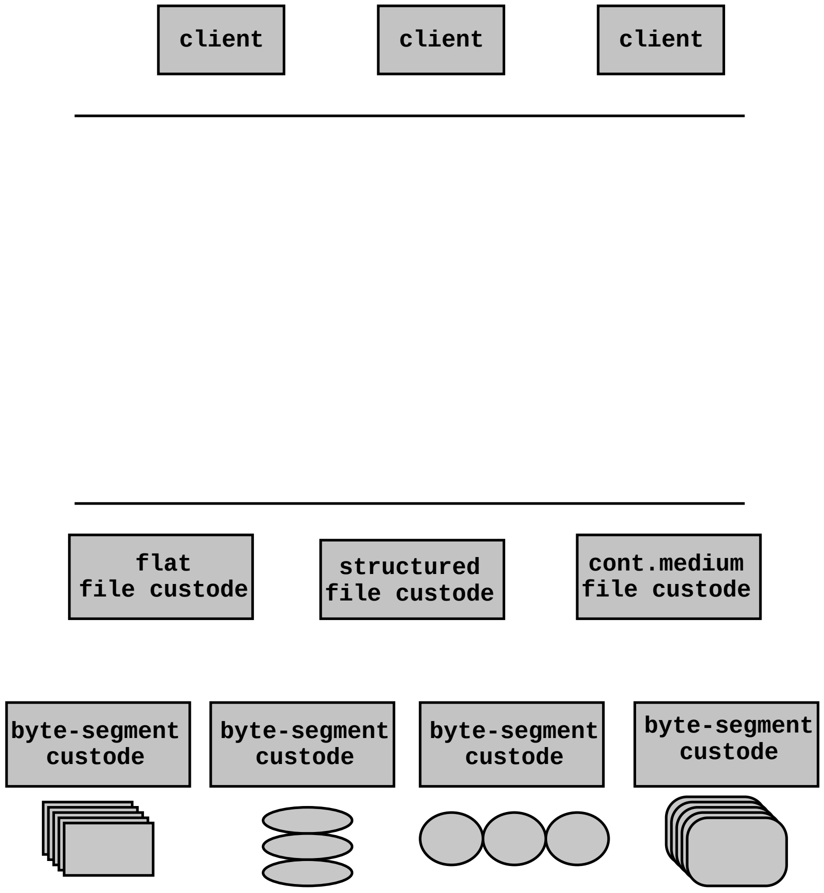
- **NVRAM used in storage substrate**

  high performance atomic update

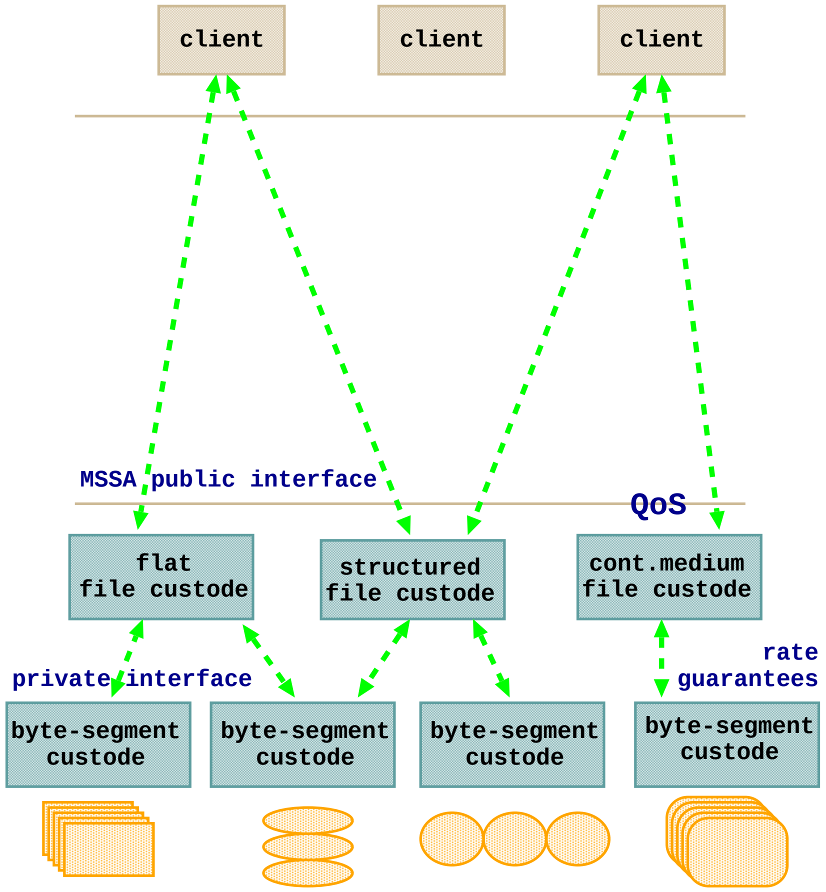
- **Value-adding service layers above MSSA**

  MSSA access control extends
  indefinitely through the layers

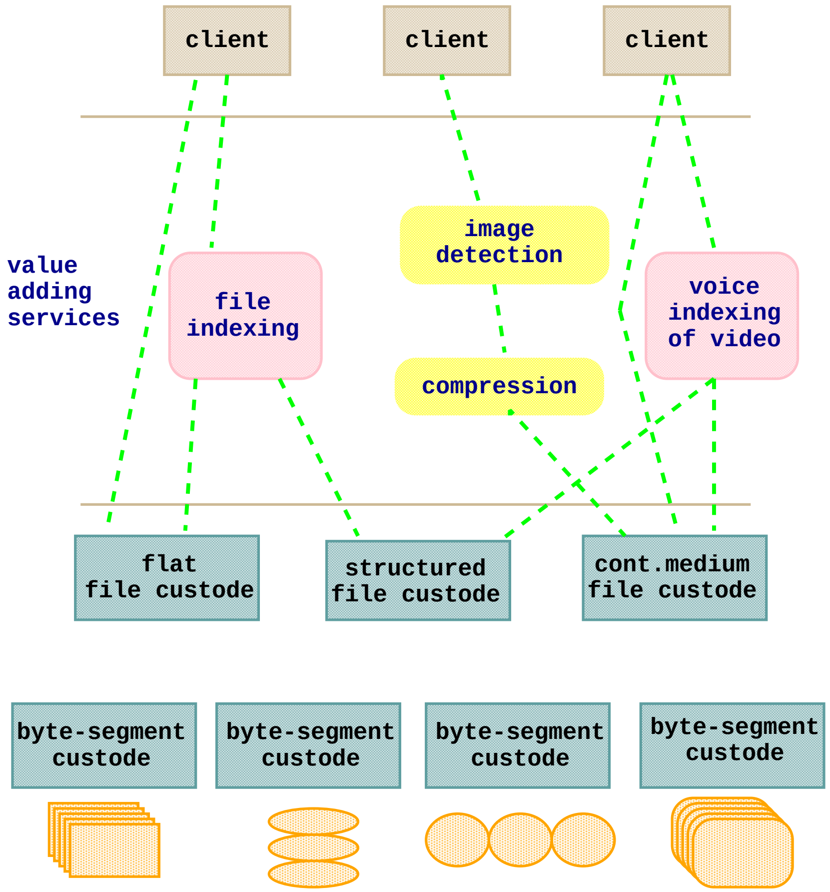
MSSA: Multi Service Storage Architecture



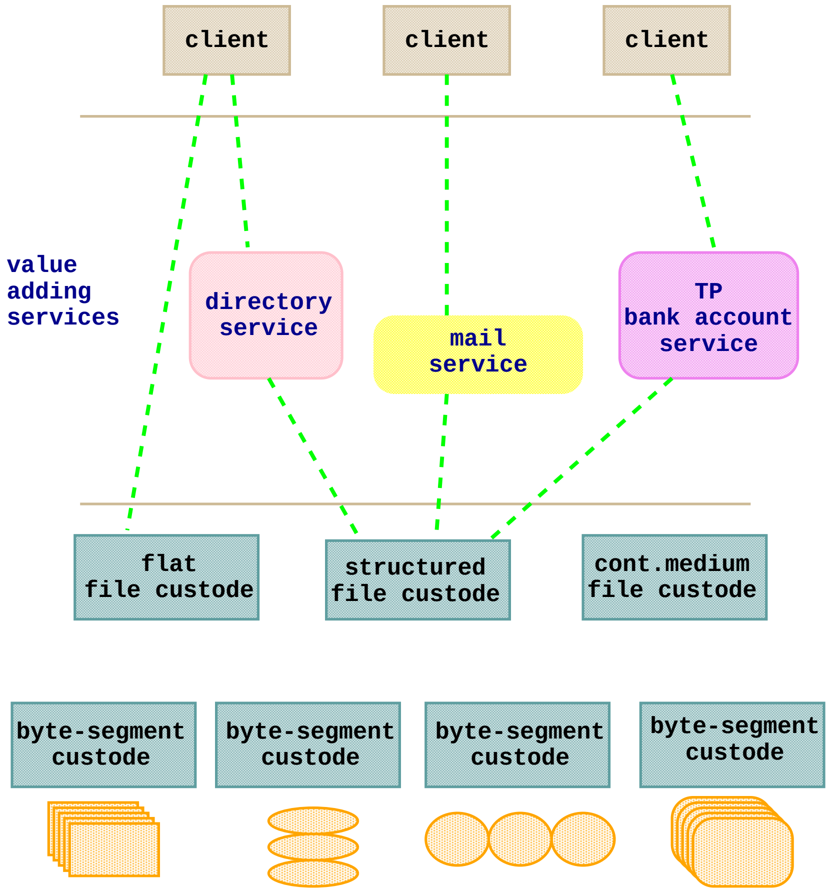
MSSA: Multi Service Storage Architecture



MSSA: Multi Service Storage Architecture



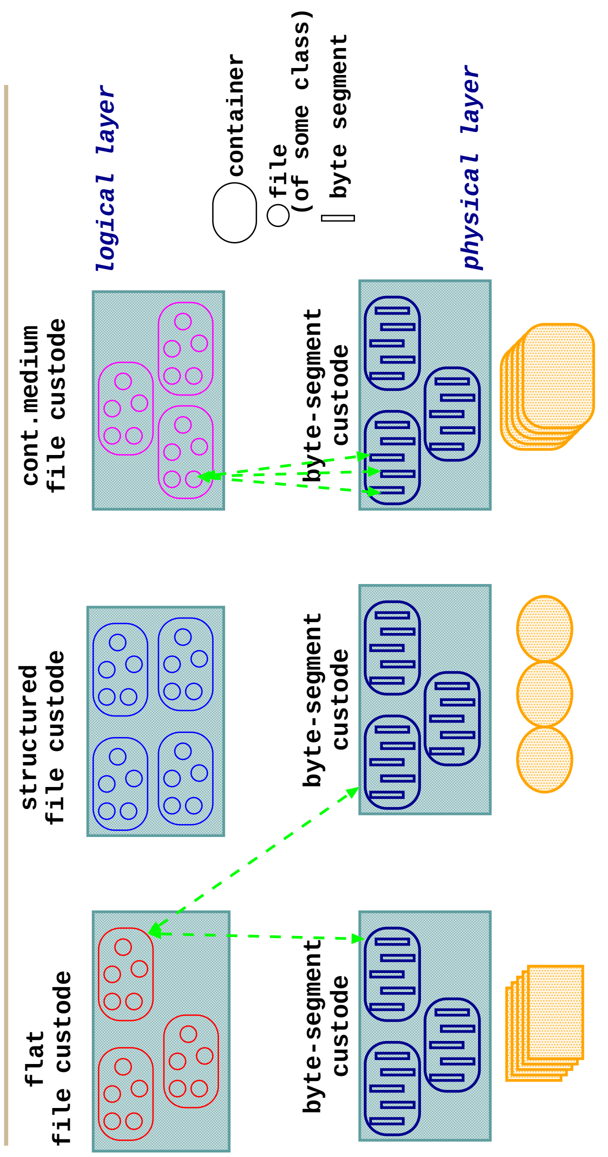
MSSA: Multi Service Storage Architecture



MSSA: Naming

- Any naming trees are external to the MSSA
- MSSA names are bit patterns: SSIDs
- SSIDs may be stored persistently e.g. in structured files
- SSID = container-name, object-name
  (32) (64)
  - unique within the object's class
  - never reused
- A single object can therefore be moved to another container
- optimisation: can often reuse the byte-segment instance number at higher levels

MSSA Files, byte segments and containers



STRUCTURED FILE CUSTODE (SFC)

Idea abstract out storage requirements common to emerging applications

Not a (language) type manager

But storage structure sufficient to locate SSIDs for existence control

(Sue Thomson PhD thesis 1990)

base types byte sequence

SSID

generators

```
sequence   T*
record     (T1,T2,...Tk)
union      [T1+T2+.. +Tk]

set ?
bag ?
```

components may vary in length

storage type-structure may be changed to support structured documents

MSSA: Multi Service Storage Architecture