# Purpose

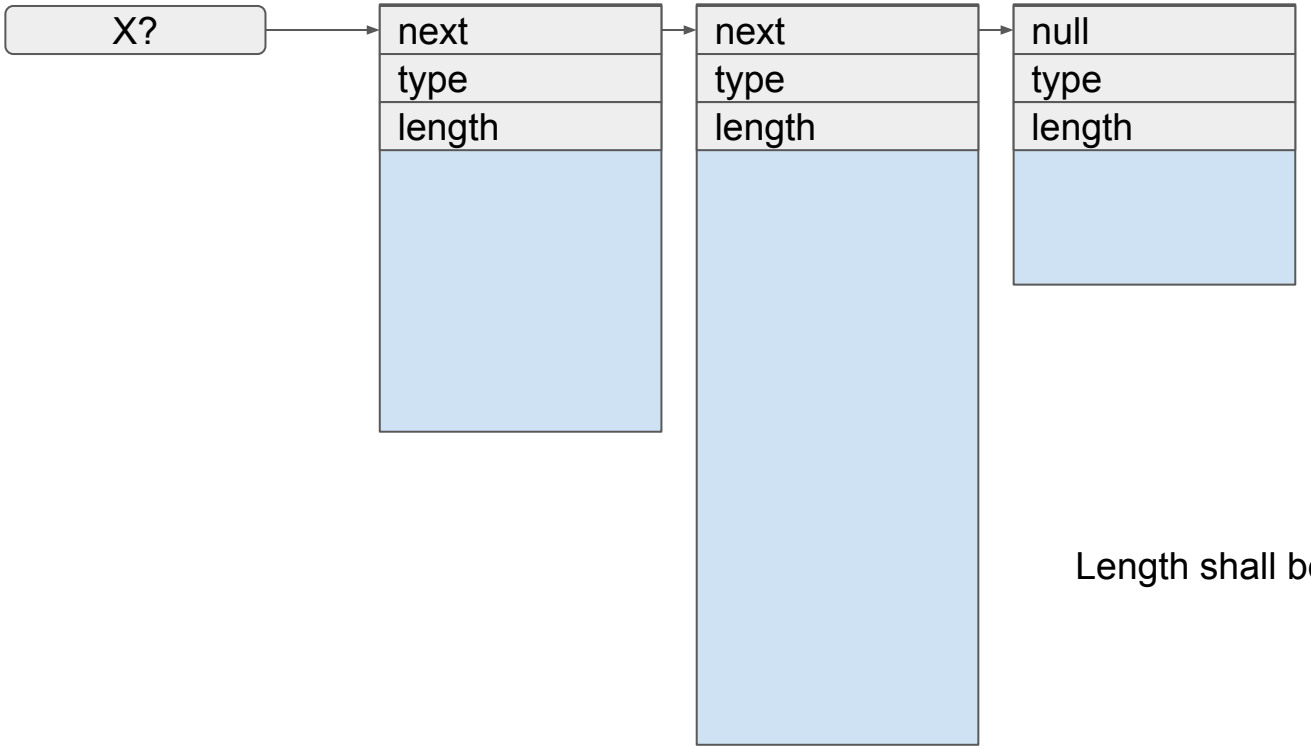
Convey dynamic “sensing” through all projects

- DIMM population information
- May not be just DRAM: NVDIMM, pure flash (Diablo technologies) and future accelerators

Convey static configuration from **authoritative** component to others

- Secure DRAM size for instance
- Beyond compile time
- Allows flexibility in the software supply chain (not all FW come from a single provider)

# Chained list (pointed to by X?)




Length shall be > 16bits

# An evaluation of metadata costs

Two DIMMs: 2 * (sizeof(address) + sizeof(length) + sizeof(spd))

C structure without metadata for versioning…

- 4 + 4 + 288 = 296 bytes

DT

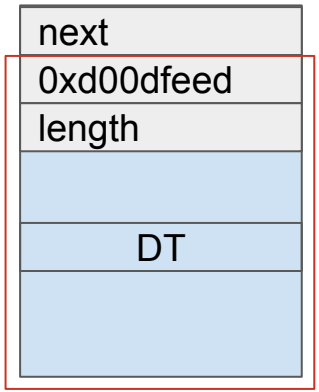
- 4 + 4 + 32 + 444 = 484 bytes

```
/dts-v1/;
/ {
    node0 {
        channel0 {
            dimm0 {
                reg = <0 0 0 0xFFFFFFFF>;
                spd = <0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 >;
            };
            dimm1 {
                reg = <0 0x80000000 0 0x3FFFFFFF>;
                spd = <0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 >;
            };
        };
    };
};
```

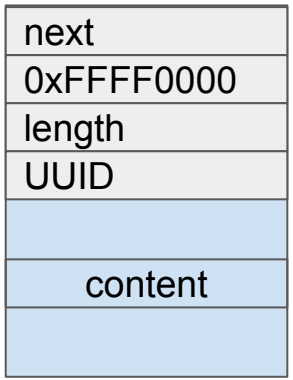
# ID tricks

If we say type is 32 bits and length 32 bits, we can assign:

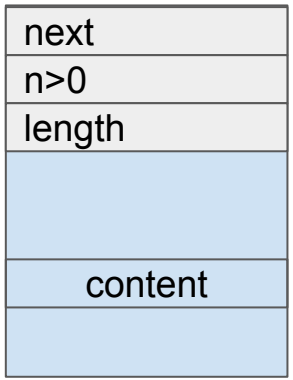
- 0xd00dfeed for DT “ID” as the DT header is “Magic” + 32 bits length
  - DT header directly continues after length.
- 0xFFFF0000 for UUID, UUID would be directly following the length.




**The red thing is a FDT format**




**UUID based HOB**




**ID based HOB**