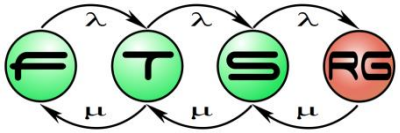
Kiberfizikai rendszerek

Data Distribution Service (DDS)

Huszerl Gábor, BME MIT



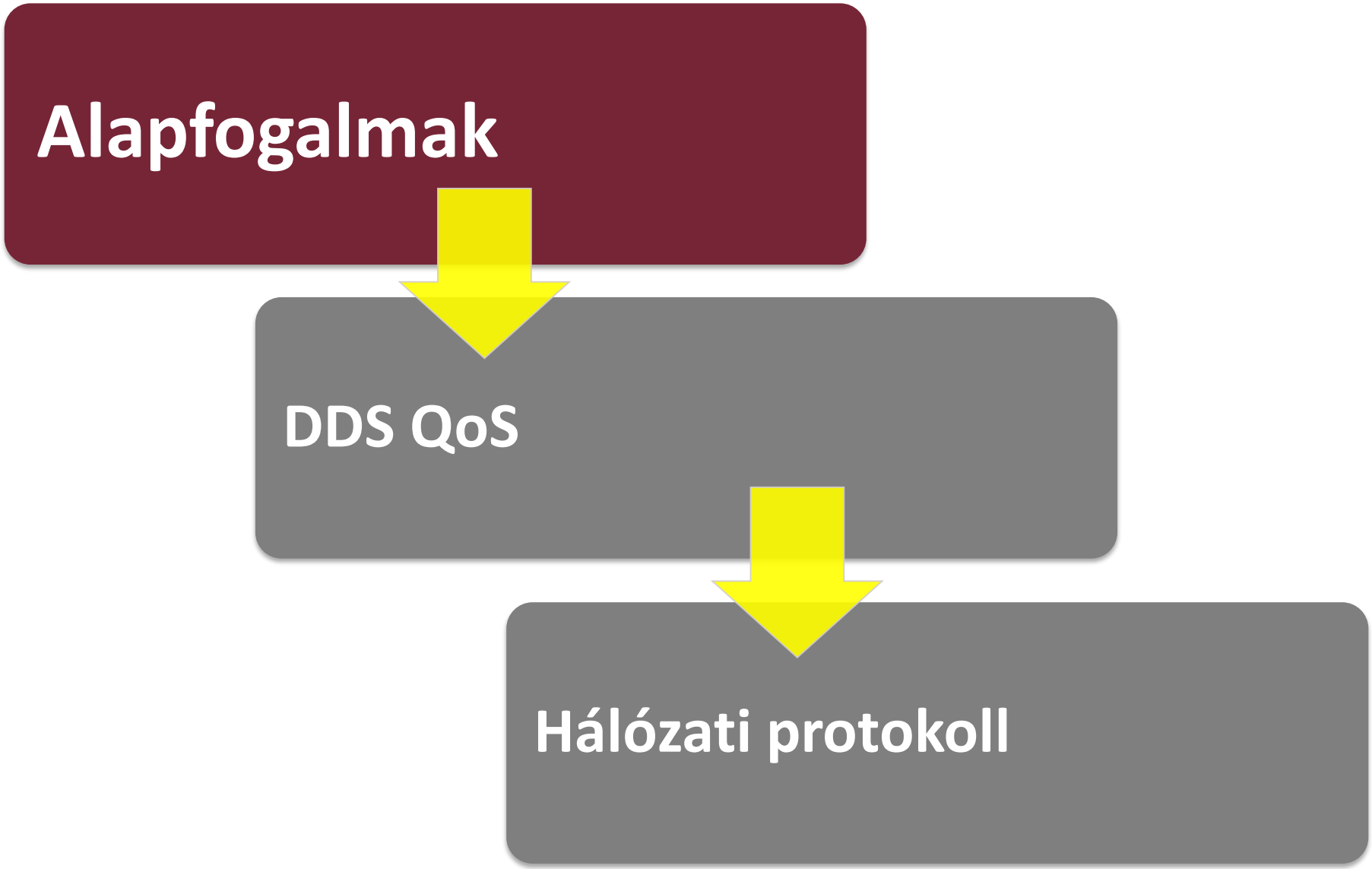
Data Distribution Service

- OMG szabvány
  - Data-Distribution Service for Real-Time Systems (DDS)
  - https://www.omg.org/spec/dds/
  - https://www.omg.org/omg-dds-portal/
- Az *IIRA* és *IICF* fontos eleme
  - Industrial Internet Reference Architecture
    https://www.iiconsortium.org/IIRA.htm
  - Industrial Internet of Things Connectivity Framework
    https://www.iiconsortium.org/IICF.htm
  - az IIOT (industrial IoT) fontos eleme

Jellemzők

- Skálázhatóság
  - rugalmasság
- Real-time
  - high-performane
  - késleltetések: 25-30 µsec, throughput>10G …
- Interoperabilitás
- Platformfüggetlenség
- Polyglot
  - Ada, C, C++, C#, .Net, Java, JavaScript, Scala, Lua, Ruby

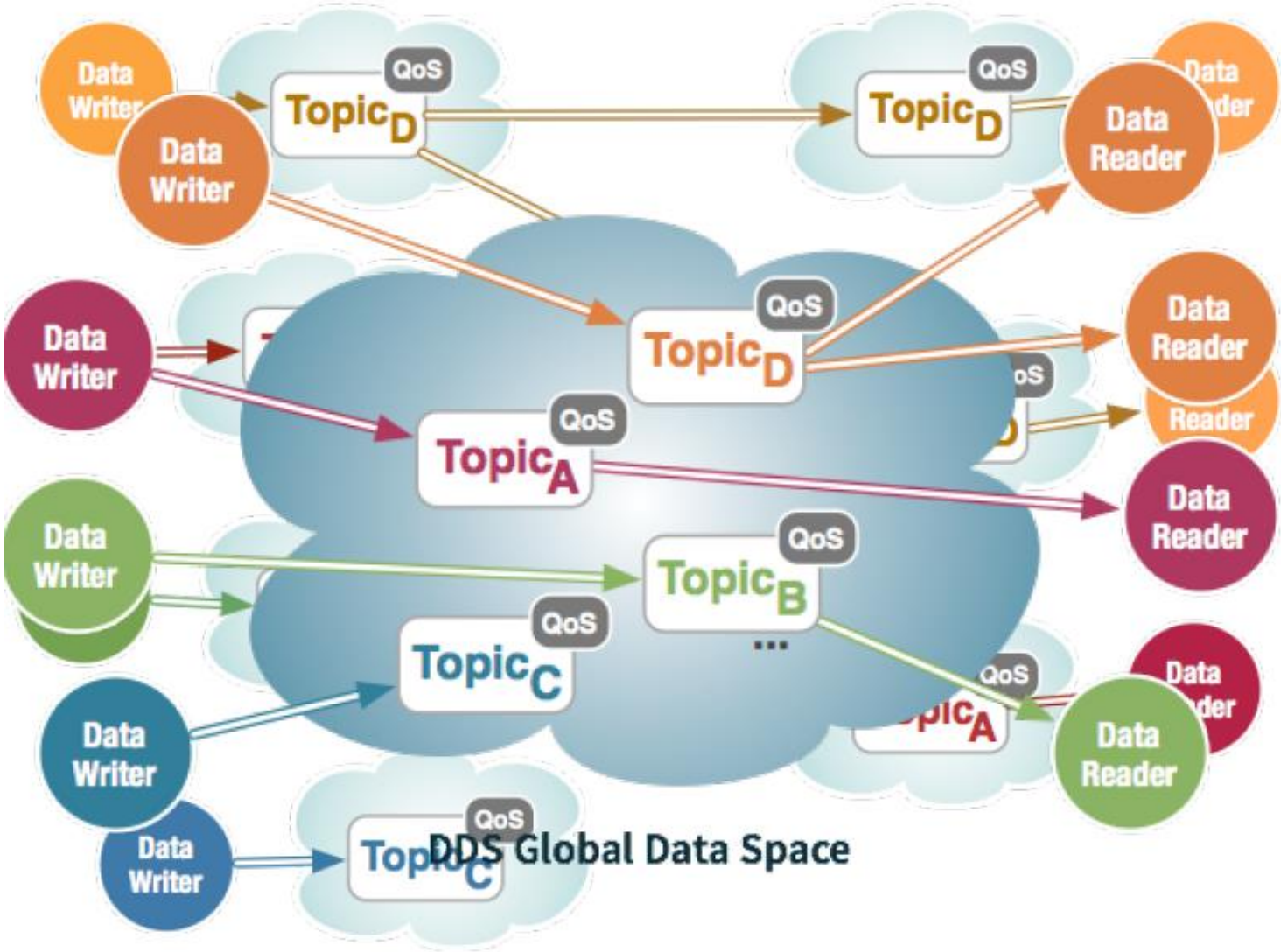
Data Distribution Service (DDS)



A koncepció

- Adatbusz
  - adatorientált együttműködés
  - adatmegosztás
- Globális decentralizált adattér
  - rendszerkomponensek közös adattere
    - lokális cache-ekkel
    - ezek kezelésének minősége (QoS) kritikus
  - közös változók írhatóak, olvashatóak
  - adatok
    - típusok, nevek, marshalling/demarshalling, láthatóság
    - típusdefiníciók (interfészdefiníciók mintájára)

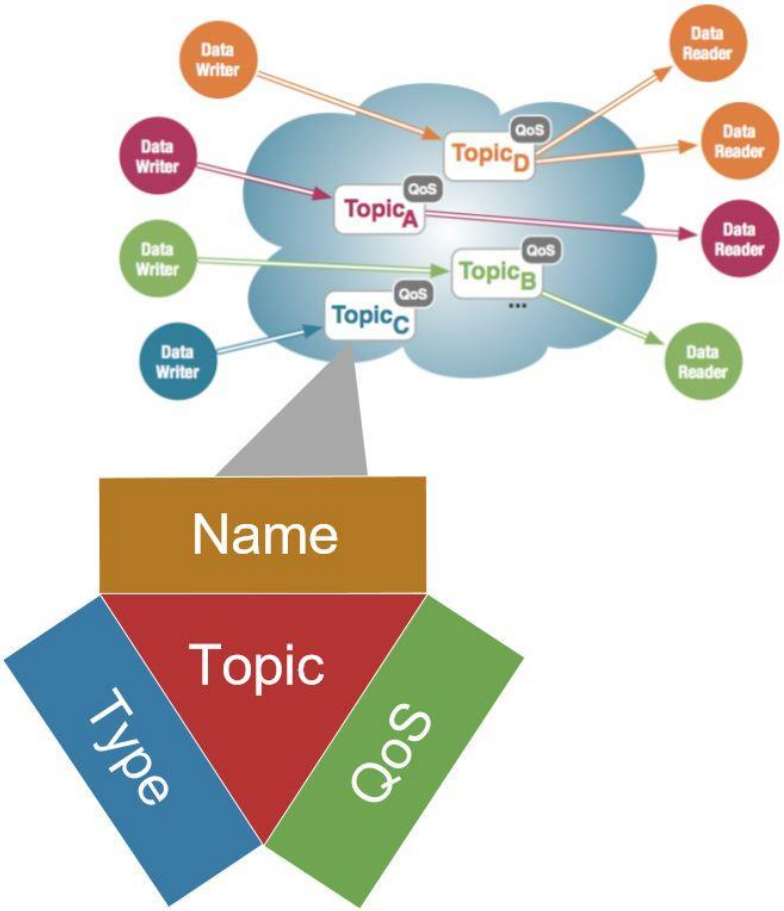
Decentralizált adattér

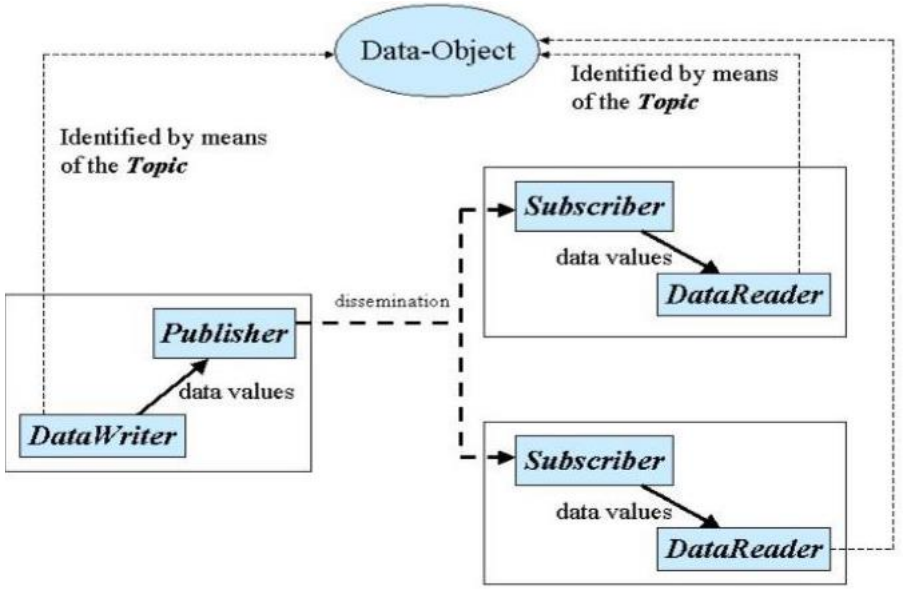


OMG DDS alapfogalmak

- **Domain**
  - alkalmazások (logikai) hálózata: közös adatbusz
  - numerikus ID azonosítja (portszámok!)
  - külön kommunikációs objektumok (DomainParticipants)
- **Topic**
  - változók (adatobjektumok) kollekciója
    - „named streams of data of the same data type”
  - egy domainen belül
  - név – típus – QoS azonosítja

OMG DDS alapfogalmak



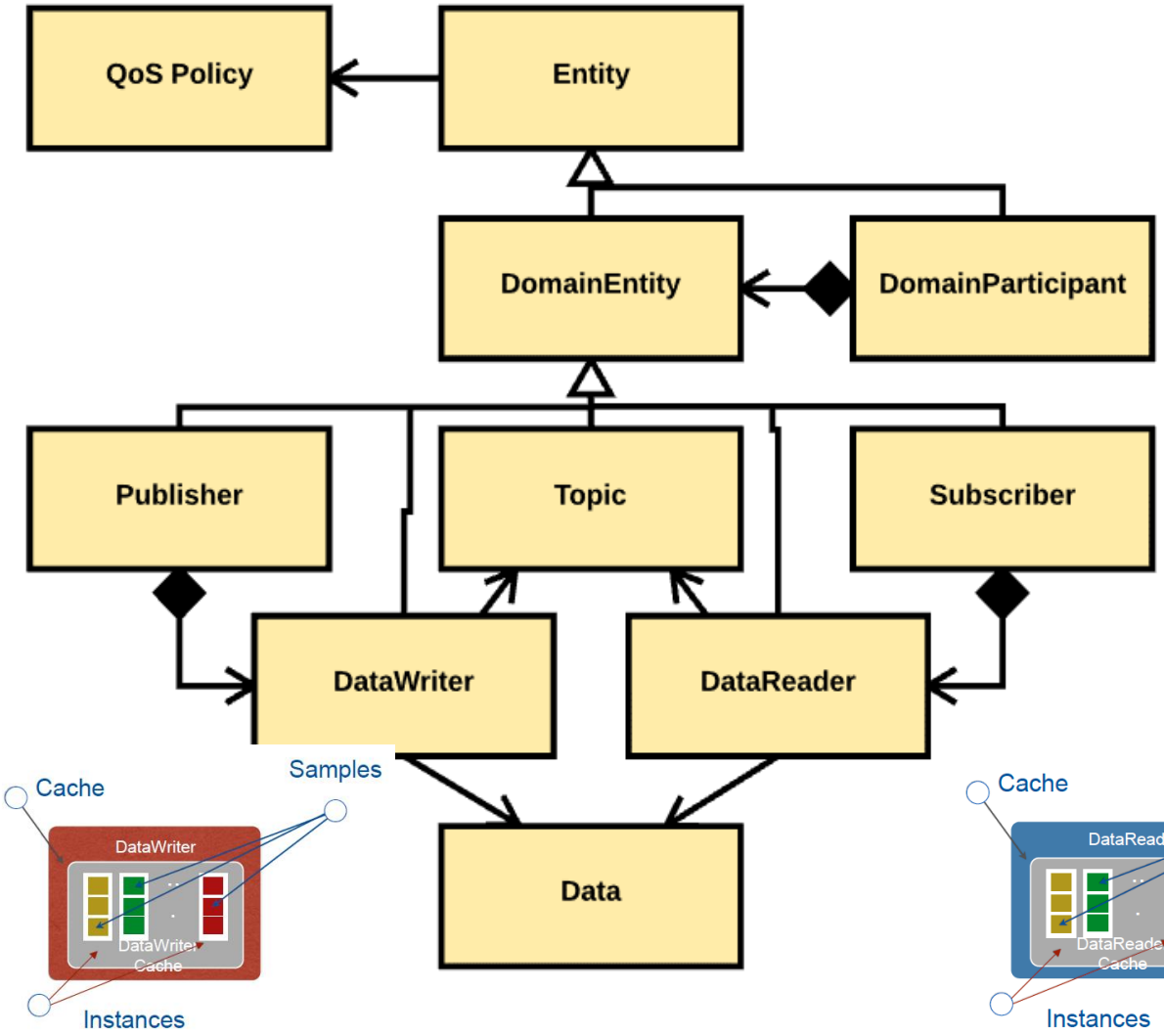


Introducing DDS™ – The Proven Data Connectivity Standard for IIoT™

OMG DDS alapfogalmak

- **Publisher/Subscriber**
  - lokális cache kezelése az alkalmazás számára
  - *Topic*-hoz kötött (adat típusa, neve kötött)
  - kapcsolat tartás, adatcsere az adott topic más *Publisher*eivel/*Subscriber*eivel
  - *Subscriber* feliratkozhat *filteredTopic*-ra is
- **DataWriter/DataReader**
  - adatok írásának/olvasásának végrehajtása az alkalmazás számára
  - *Publisher*hez/*Subscriber*hez kötött

OMG DDS alapfogalmak



OMG DDS alapfogalmak

- A *Topic* meghatároz egy adattípust (osztály)
- Ennek a típusnak lehet
  - egyetlen példánya
  - több példánya
- Kulcs attribútum
  - az adattípus példányainak megkülönböztetésére
  - nincs kulcs ⇒ egyetlen példány van (singleton)
- Az éppen publikált adat lehet
  - meglévő példány (**instance**) újabb értéke (**sample**)
  - új példány első értéke (kulcs attribútum új értékével)

OMG DDS alapfogalmak

- Adatpéldányok (instances) állapotai
  - ALIVE
  - NOT_ALIVE_NO_WRITERS (nincs élő DataWriter hozzá)
  - NOT_ALIVE_DISPOSED (nincsen szükség rá többé, törölhető)
- Értékek (samples) állapotai
  - READ, NOT_READ
- Olvasás (views) állapotai
  - NEW (az első olvasás, amióta a példány most ALIVE)
  - NOT_NEW (ezt a példányt már láttuk korábban)
  - minden *DataReader*re, minden példányra külön

Adattípusok leírása

- Interfészleíró fájl
  - IDL fájl (OMG CORBA szabványból ismert)
  - XML fájl
- Kódgenerátor (célnyelvenként)
  - `Struct VideoStream ⇒`

```
VideoStream.java
VideoStreamDataReader.java
VideoStreamDataWriter.java
VideoStreamSeq.java
VideoStreamTypeCode.java
VideoStreamTypeSupport.java
```

Data Structure Definition (IDL)

```
const long MAX_BUFFER_SIZE = 1048576;

struct VideoStream {

        // This allows a subscriber to differentiate between different video
        // publishers that are publishing the same data on the same Topic.
        long stream_id; //@key

        // Some video formats may require metadata to be sent with the binary
        // video data, such as flags or a frame sequence_number.
        unsigned long flags;
        unsigned long sequence_number;

        // This contains the video buffers
        sequence <octet, MAX_BUFFER_SIZE> frame;
};
```

Adattípusok leírása

- DDS által közvetlenül kezelhető típusok:
  - structure, union, value type
- Ezekbe ágyazható egyéb típusok:
  - enum, type def
  - primitív típusok:
    - octet, char, wchar
    - short, unsigned short
    - long, unsigned long, long long, unsigned long long
    - float, double, long double
    - boolean, enum (explicit értékekkel vagy azok nélkül)
    - bounded/unbounded string/wstring

Adattípusok leírása

- További támogatott konstrukciók:
  - modul (package, namespace)
  - mutató
  - primitív vagy felhasználói típusok tömbje
  - láncolt lista
  - bitfield
  - typedef, union, struct, value type

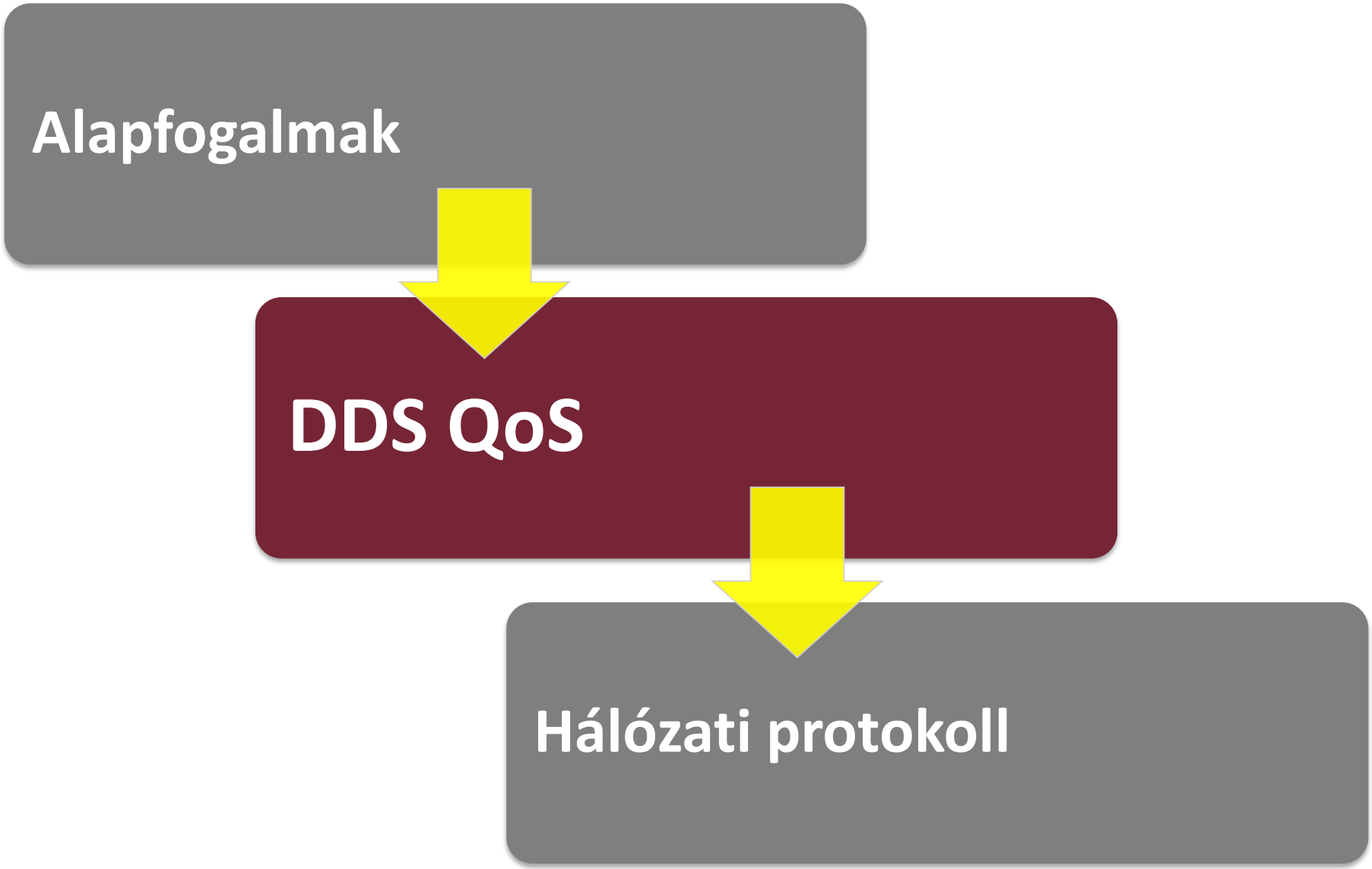
Szinkron vs. aszinkron kommunikáció

- Szinkron: write, read, take (read/take nem blokkol)
  - read/take_instance, read/take_next_instance, read/take_next_sample, read/take_w_condition, …
  - feltétel lehet állapot alapú vagy tartalom alapú
- Szinkron: waitSet
  - várakozás (blokkolás), amíg a feltétele teljesül
    - pl. van elérhető új adat
    - pl. van megfelelő tartalmú adat
- Aszinkron: Listener
  - call back függvény a DDS eseményeinek kezelésére
    - függvényt megírni, megfelelő helyen regisztrálni, DDS hívja

Dinamikus felderítése

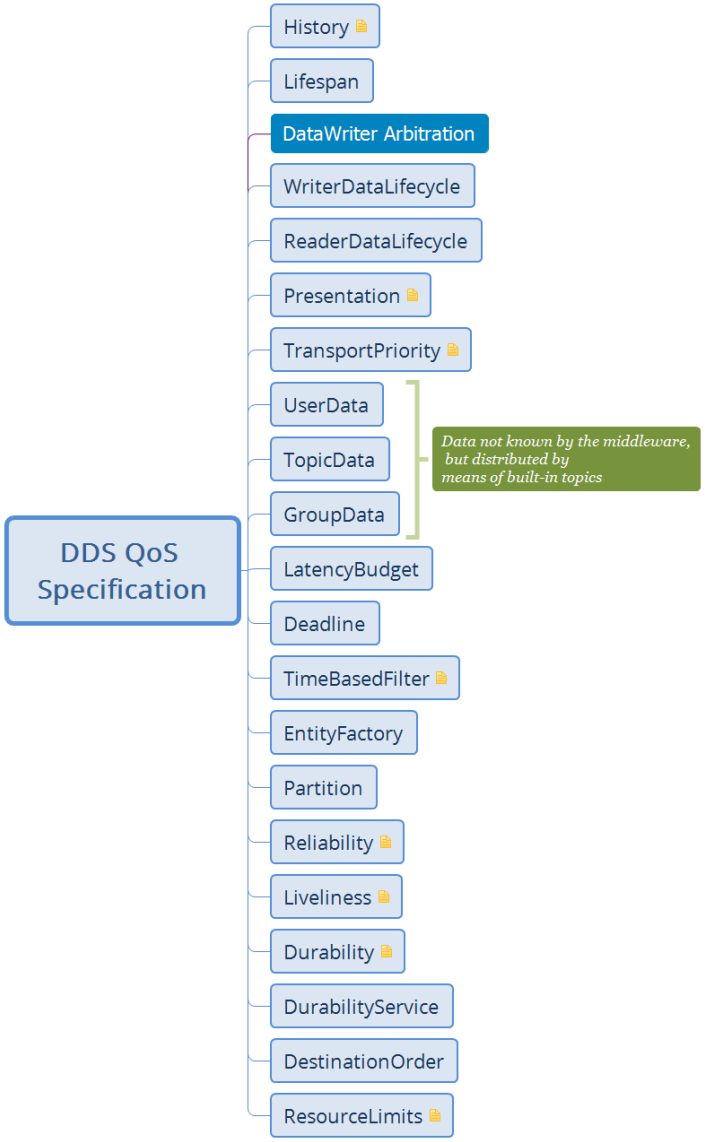
- Futás közbeni felderítés (plug and play)
  - Ki van a *Domain*ben? (Milyen IP címmel?)
  - Milyen adatot ír/olvas? (*Topic*, adattípus, változónév)
  - Milyen QoS-sel?
- LAN-on belül
  - LAN-wide multicast
- LAN-on túl
  - *initial peers list*
    - unicast, multicast címek
    - konfigurációból olvasható
    - futás időben bővíthető (kódból és a többiektől)

Data Distribution Service (DDS)

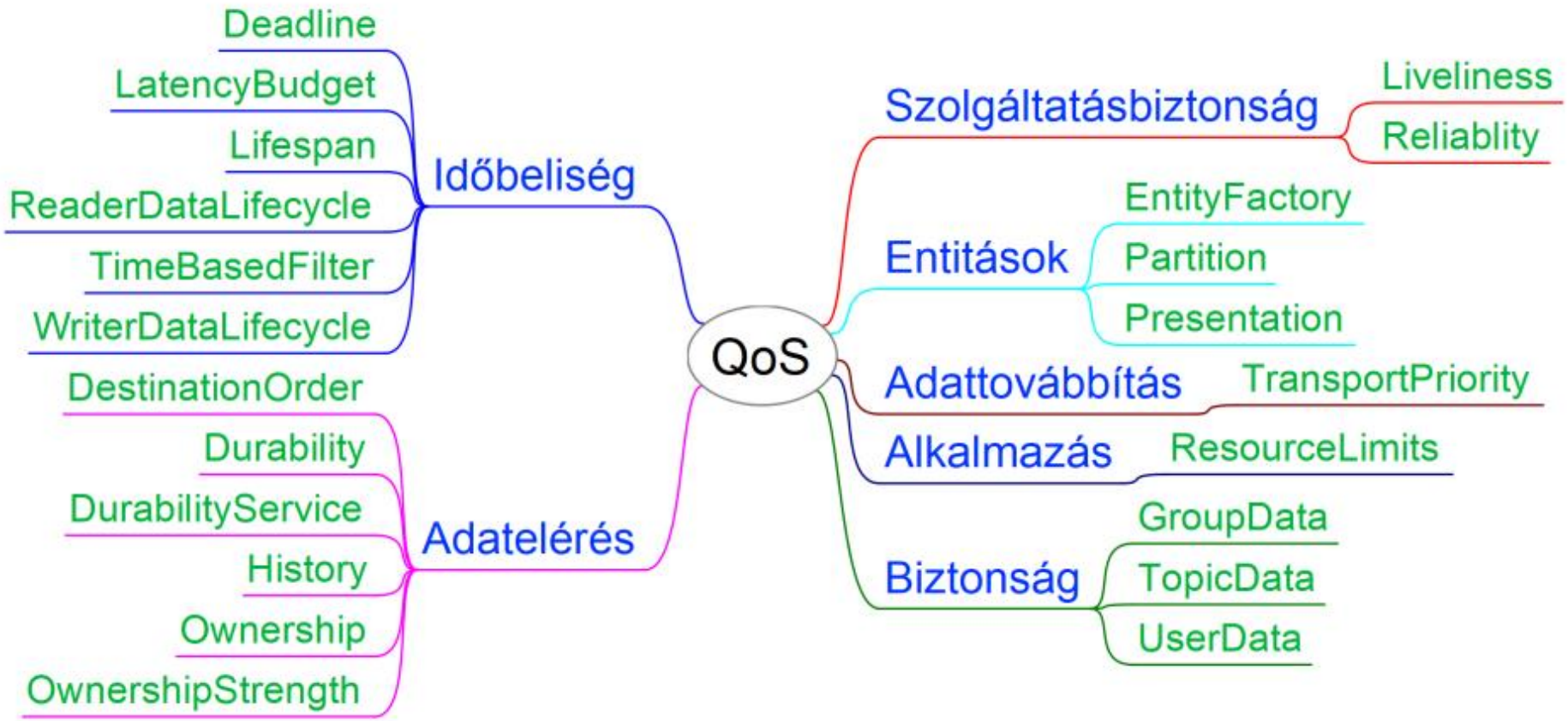


DDS QoS

- Quality of Service
  - requested vs. offered
  - bármely *entity*-nek lehet saját, ö
  - kommunikáció feltétele a kompa
    - Példa: megbízhatóság
      „best effort” publisher és
      „reliable” subscriber kompatibilis,
      fordítva nem
- Beállítható
  - konfig. fájlból (XML)
  - programozottan, futás időben is



QoS-es aspektusok szerint



DDS QoS (megbízhatóság)

- QoS: Reliability (kézbesítés garantált-e?)
  - BEST_EFFORT, RELIABLE
- QoS: History (hány érték tárolandó?)
  - KEEP_LAST, KEEP_ALL (első esetben mélység beállítandó)
- QoS: Durability
  - VOLATILE (history nincs mentve)
  - TRANSIENT (history mentve a memóriába) erőforráskorlátok betartása!
  - PERSISTENT (history mentve perzisztensen)

DDS QoS (idő)

- QoS: Deadline (mennyi időnként jön adat?)
  - író oldal ígérete
  - olvasó oldal kivételt dob, ha túl sokáig nem jön adat
- QoS: Liveliness („heart beat” üzenetek)
  - típusa: automatikus/kézi (middleware/alkalmazás által kezelt)
  - olvasó oldal kivételt dob, ha kihagy
- QoS: Time_Based_Filter (ne jöjjön túl sűrűn adat)
  - az olvasó alkalmazást védi értékek eldobásával

DDS QoS (kézbesítési sorrend)

- QoS: Presentation
  - access_scope
    - INSTANCE: példányon belüli függőségek
    - TOPIC: egy *Topic* összes példánya feletti függőségek
    - GROUP: egyetlen *Publisher* összes *DataWritere* felett
    - HIGHEST_OFFERED: amit a *Publisher*ek tudnak
  - ordered_access
    - DataReader queue-ja tükrözi a küldés sorrendjét (scope-on belül)
  - coherent_access
    - csoportos kézbesítés
    - publikáló oldalon `begin_coherent_changes()`, `end_coherent_changes()` kell

DDS QoS (kézbesítési sorrend)

- QoS: Destination_Order
  - különböző *DataWriter*ek publikálásainak kezelése
  - *DataReader*enként beállítható (konzisztencia?)
  - BY_RECEPTION_TIMESTAMP
    - megfelelő Ownership_Strength esetén elegendő lehet
  - BY_SOURCE_TIMESTAMP
    - azonos Ownership_Strength-ek esetén kellhet
    - késve érkező értékeket *Subscriber* eldob
    - óraszinkronizálás?

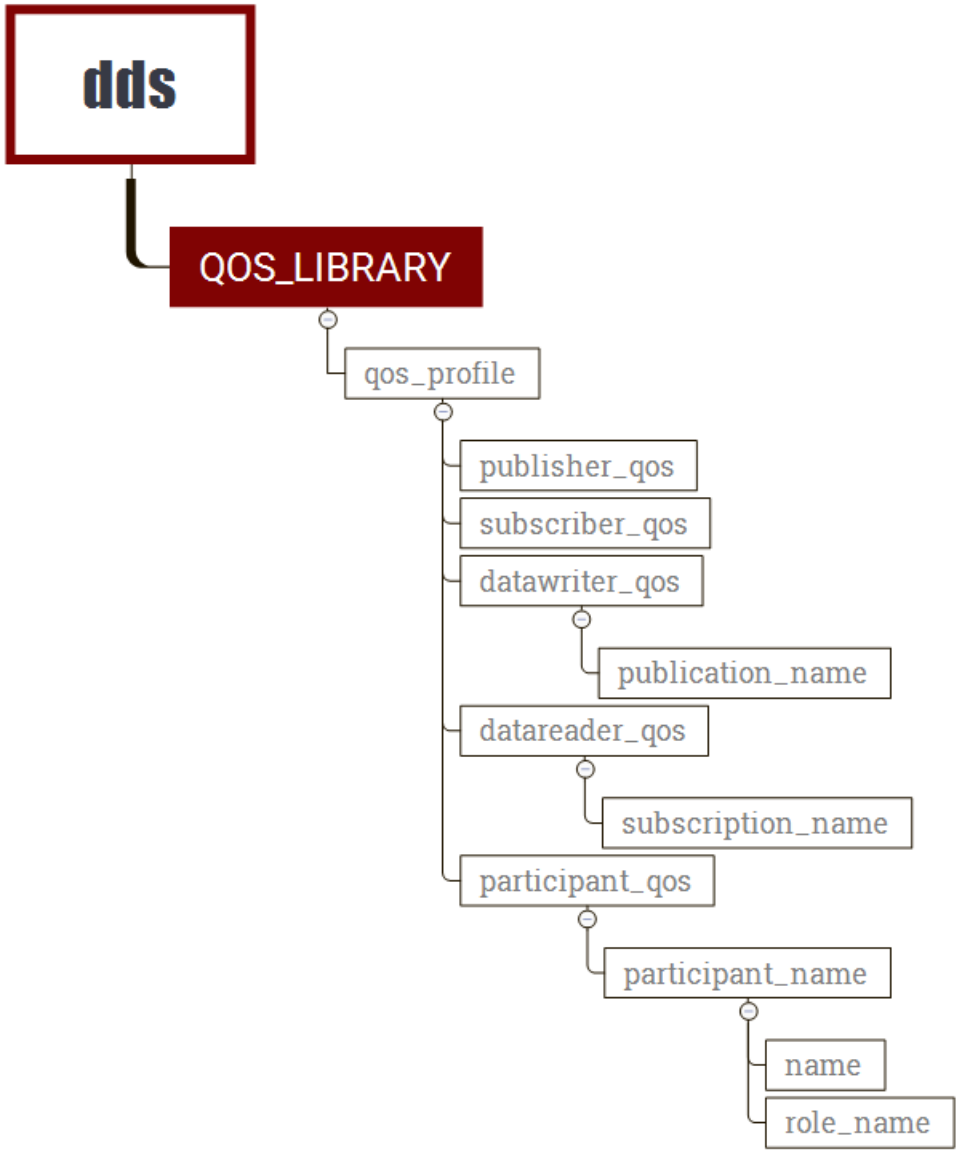
DDS QoS (erőforrás gazdálkodás)

- QoS: Resource_Limits
  - max_samples (összesen)
  - max_samples_per_instance, max_instances
- QoS: Latency_Budget
  - RT alkalmazásoknál össz. átviteli késleltetés korlátja
  - késleltetés: küldő oldali (DDS, UDP, IP, …, fizikai) + hálózati + fogadó oldali (fizikai, …, IP, UDP, DDS)

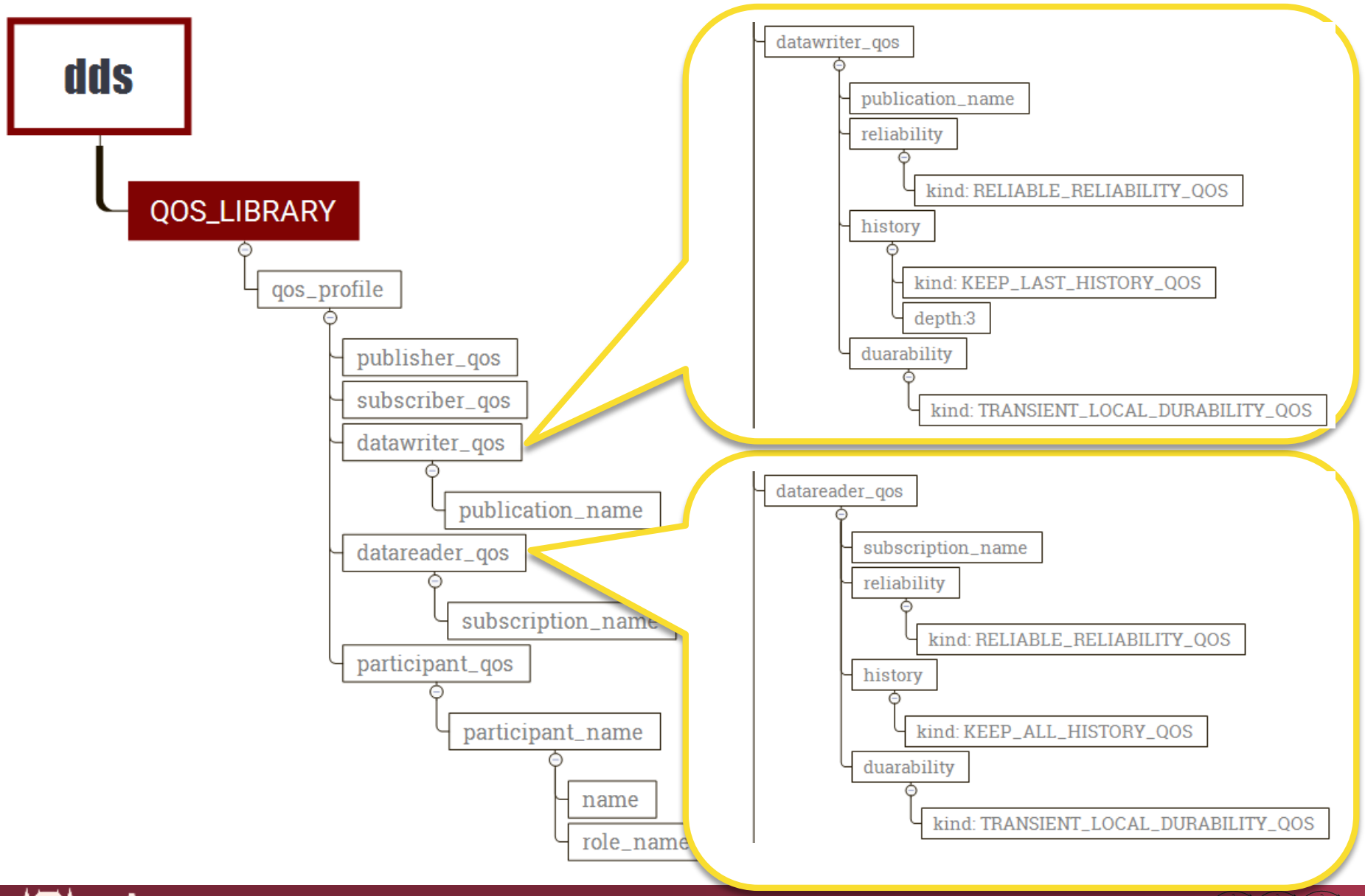
DDS QoS (egyéb)

- QoS: Partition
  - *Topic*-on belüli logikai bontás, nem ad biztonságot
  - wildcard karakterekkel (offered és requested kompatibilitása!)
- QoS: Ownership_Strength (integer)
  - írók birkózása az adatpéldány írási jogáért
  - a legerősebb élő írhatja a példányt („melegtartalék”)
  - csak olyan Topic-oknál, ahol „Ownership = Exclusive”
- QoS: User_Data
  - üzenetek metaadatainak saját kiegészítő része
  - pl. security infók

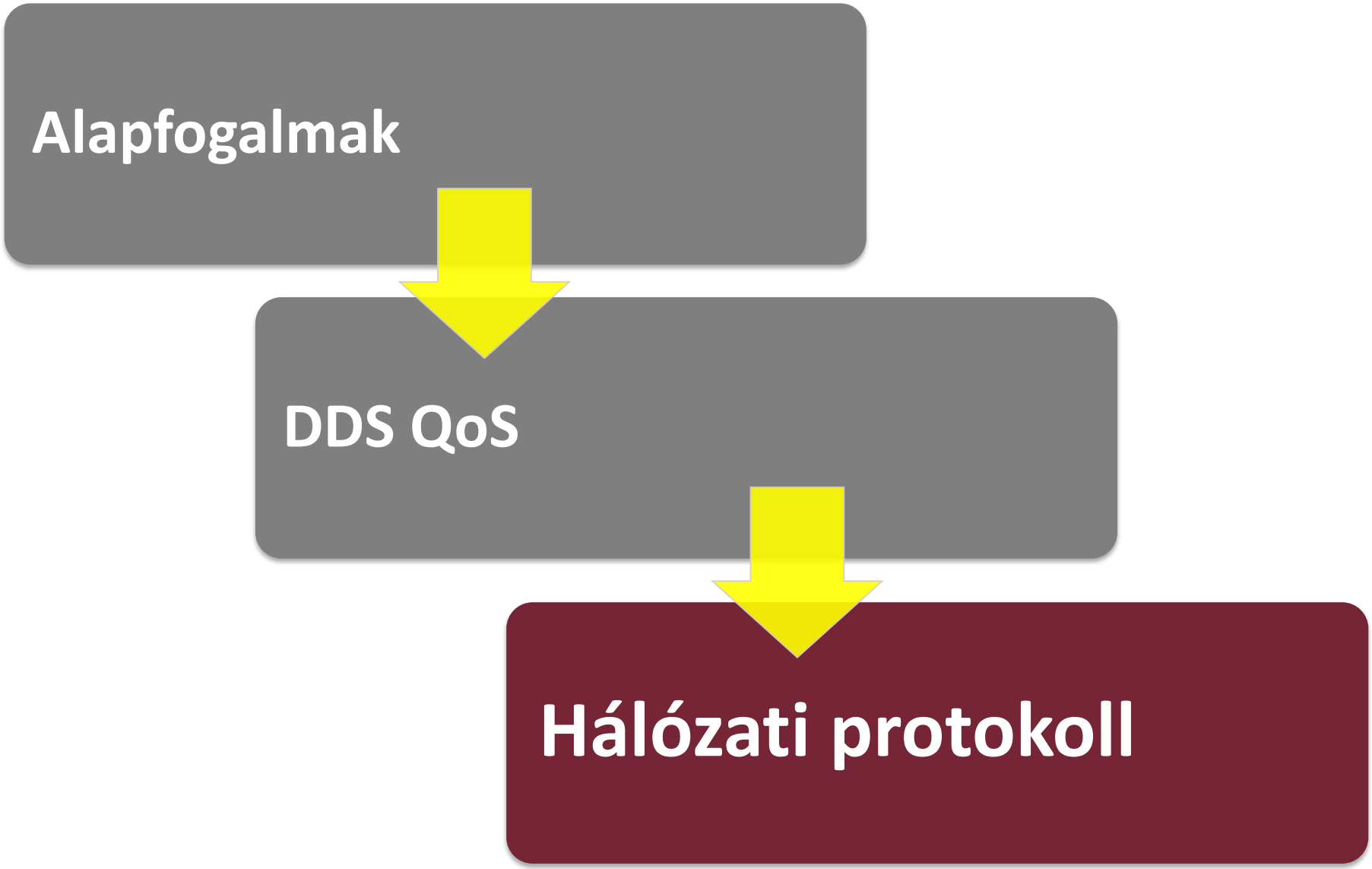
USER_QOS_PROFILES.XML



Reliability (XML)



Data Distribution Service (DDS)



Hálózati kommunikáció

- Portok:
  - discovery multicast: 7400 + (250 * Domain) + 0
  - user multicast: 7400 + (250 * Domain) + 1
  - discovery unicast: 7400 + (250 * Domain) + (2 * Participant) + 10
  - user unicast: 7400 + (250 * Domain) + (2 * Participant) + 11
  - *ephemeral ports*: „above 32768 or higher” (OS függő)

Hálózati protokoll

- RTPS (Real-Time Publish-Subscribe Protocol)
  - Real-Time Innovations, Inc. terméke
  - IEC: *Real-Time Industrial Ethernet Suite* szabvány (IEC-PAS-62030)
  - multi-vendor DDS rendszerek interoperabilitási protokollja is (https://www.omg.org/spec/DDSI-RTPS)
  - UDP/IP felett
    - megbízható kommunikáció nem megbízható protokoll felett
  - P/S protokoll
    - üzenet alapú, aszinkron kommunikációs middleware
    - publishers, subscribers, topics
    - jó 1-to-N skálázódás, gyors, nem feltétlenül megbízható

Hálózati protokoll

- RTPS kommunikációs modelljei
  - P/S protokoll: adatok átvitele
  - Composite State Transfer (CST) protokoll: állapotinformációk átvitele
- RTPS résztvevők:
  - Writers (Publications és CSTWriters)
  - Readers (Subscriptions and CSTReaders)
- RTPS funkciói
  - adatmegosztás: protokoll mindkét komm. modellhez
  - adminisztráció: protokoll a résztvevők attribútumainak megosztására (résztvevők tudhatják egymás állapotát)

Hálózati protokoll

- RTPS logikai üzenetei
  - ISSUE: „hasznos adatok” átvitele (adatmegosztás)
  - VAR: résztvevő állapotának átvitele
  - HEARTBEAT: Writer állapotának átvitele
  - GAP: információk elavultának átvitele
  - ACK: Reader állapotának átvitele
- Ki kinek mit küld?
  - Publication to Subscription(s): ISSUEs, HEARTBEATs
  - Subscription to Publication: ACKs
  - CSTWriter to CSTReader: VARs, GAPs, HEARTBEATs
  - CSTReader to CSTWriter: ACKs

Hálózati protokoll

- Adatok reprezentációja a vezetéken
  - OMG CDR (Common Data Representation) szabvány

- RTPS referencia implementációk
  - állapotmentes
    - jól skálázódik, kevés memóriát használ
    - „best effort” átvitelre különösen jó
  - állapottal rendelkező
    - többi résztvevő állapotát tárolja, nagy memória igény, kis sávszélesség igény
    - megbízható átvitelre különösen jó