MCR & DTI [523]

Design for the Distributed Data Locator Service for Multi-site Data Repositories

presented by Nakanishi H.

National Institute for Fusion Science (NIFS), NINS, Japan

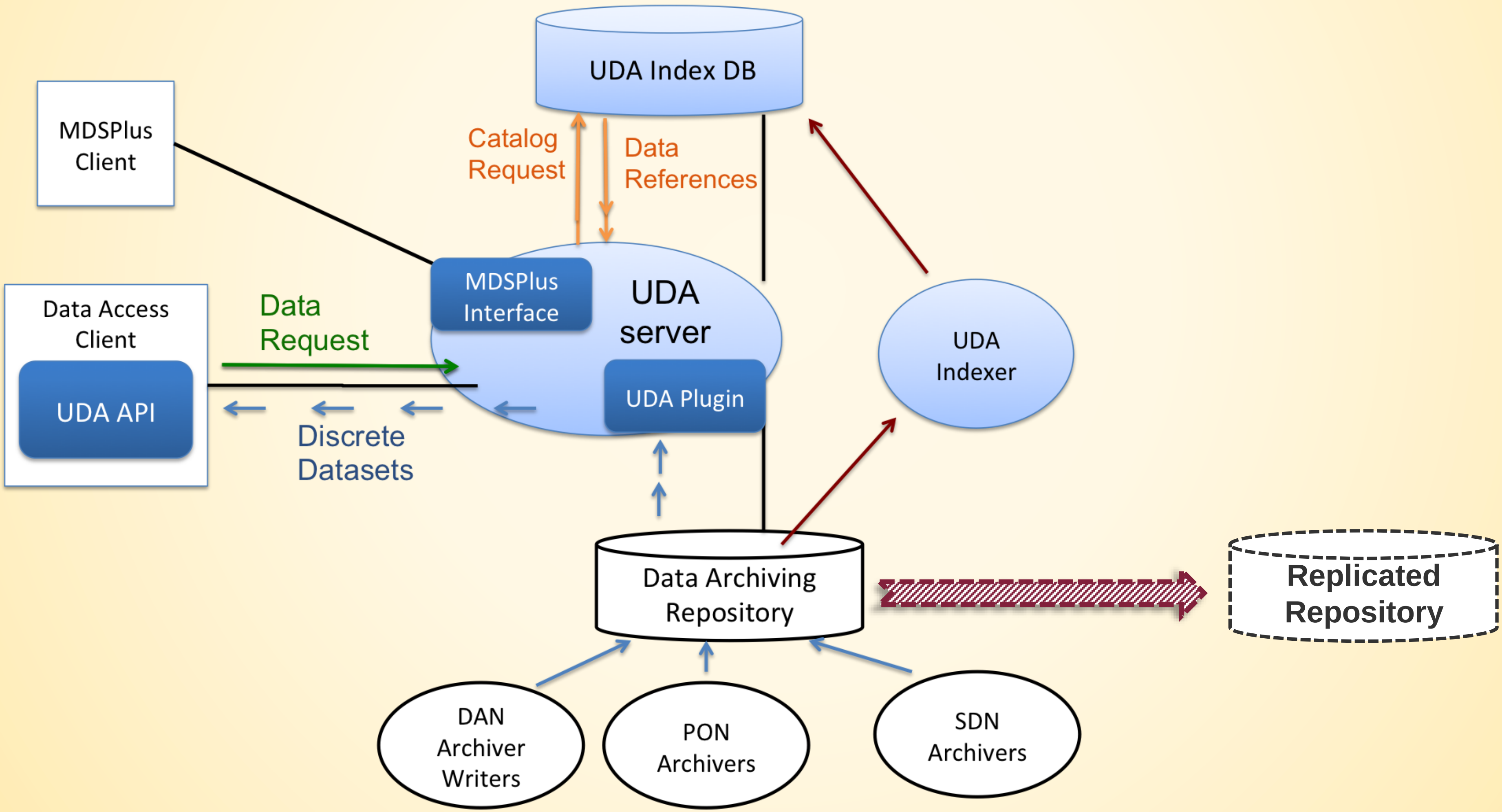
on behalf of
NIFS, NII, and QST research collaborators

Backgrounds and Objectives

- Massive data analyses need **high-bandwidth, low-latency data access** to storage.
  - ➢ Need a super-computer cluster together with a huge, local data storage
- **For ITER** huge data analyses,
  - ➢ JA-DA will prepare huge computer & storage at ITER REC in Rokkasho.
  - ➢ Inter-continental data replication method has been well tested. → *cf.* **MMCFTP**
- **Fusion Virtual Laboratory** in Japan gathers data from **LHD + 3 remote sites**.
  - ➢ FVL shares a central storage & index DB → will be a **"SPoF"** in accidents
  - ➢ Multi-tier storage can queue data at every stage, but Index DB should be always on service.
  - → Index DB must be a redundant, distributed service by using **multi-master DB**.
- In this study, bi-directional replication between multi-master index DB has been designed and tested by using the LHD data system on FVL.
  - ➢ Bi-directional replication is enabled by **"BDR extension"** module for PostgreSQL version 9.4 and higher.

ITER UDA structure

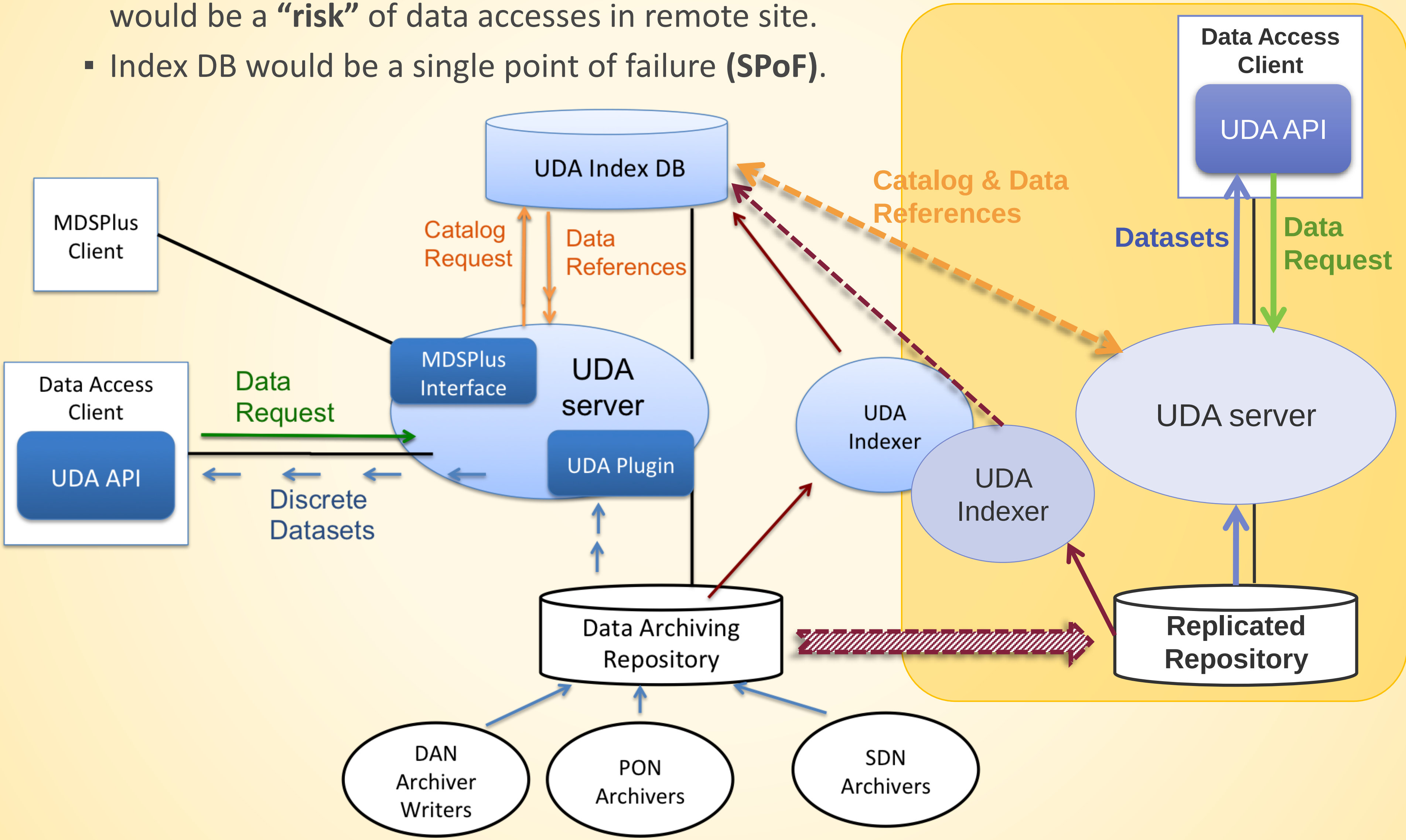
- ITER on-site data repository is a single substance. If having a replica, …



cf. UDA architecture diagram "UDA user manual" (TPLTKG v2, 2018)

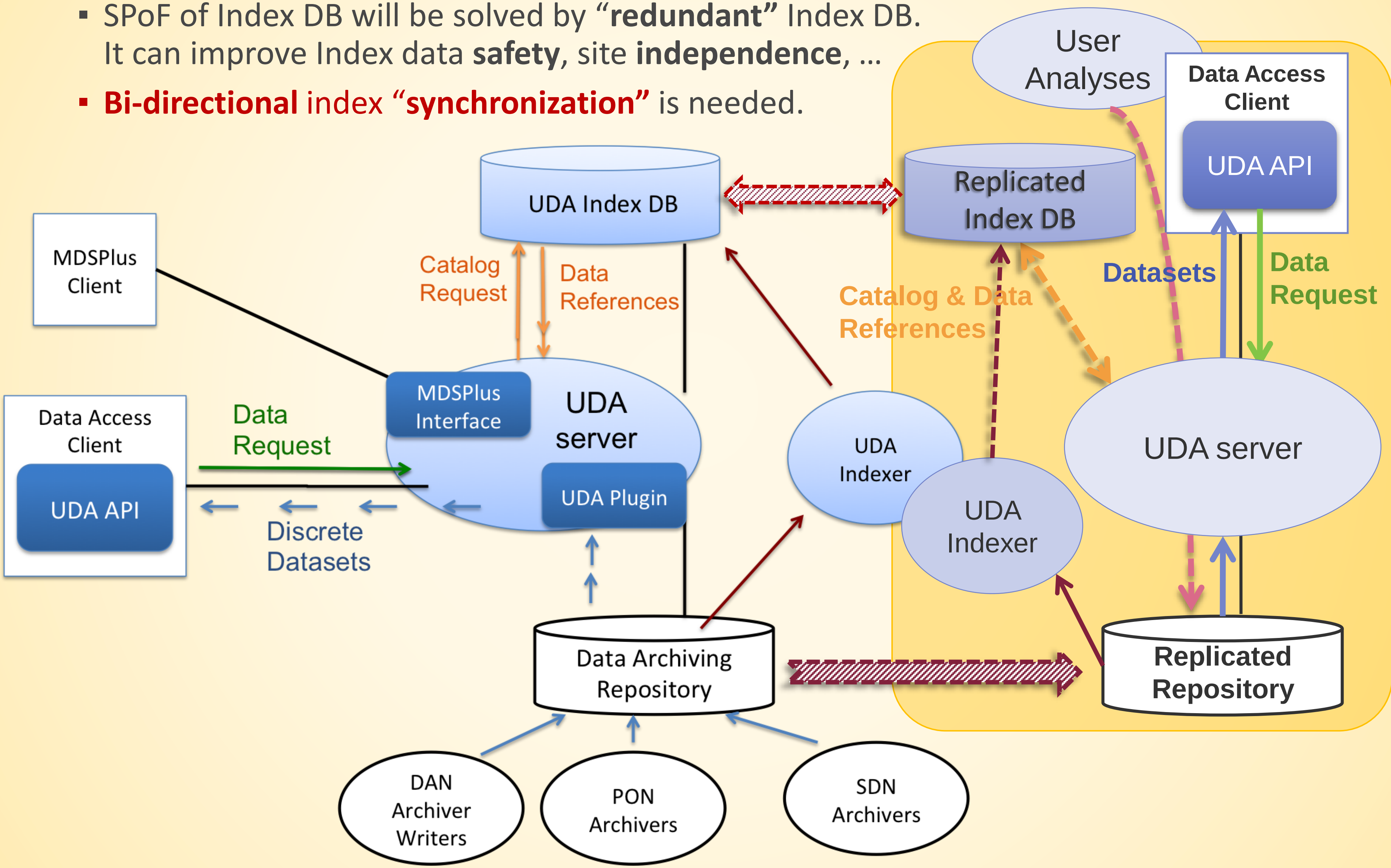
ITER UDA structure with remote repository

- Long-distance communications for accessing the Index DB would be a **"risk"** of data accesses in remote site.
- Index DB would be a single point of failure **(SPoF)**.



ITER UDA structure with "replicated index"

- SPoF of Index DB will be solved by "**redundant**" Index DB. It can improve Index data **safety**, site **independence**, …
- **Bi-directional** index "**synchronization**" is needed.

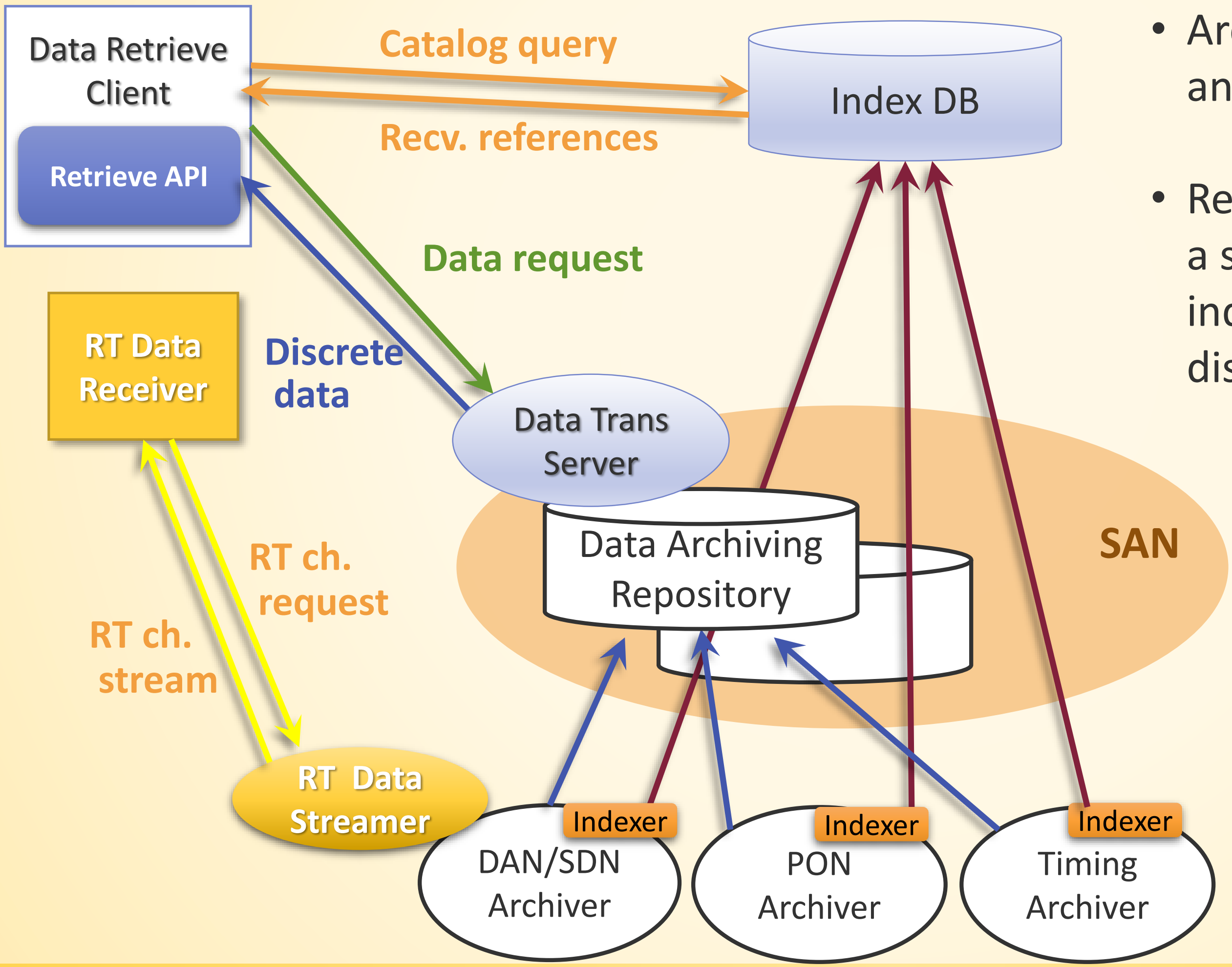


How implement ?

- **Postgres BDR** has a loosely coupled shared-nothing multi-master design.
- Bi-directional replication **(BDR)** is an extension package of PostgreSQL version 9.4 and higher.
  - ➢ can be introduced into standard PgSQL by “CREATE EXTENSION bdr” command
- As BDR is based on the PgSQL **logical replication**,
  data will be modified by **“row-based replication”**,
  neither by high-level **statement-based** nor by low-level **log-based** manners.
  - ✓ SQL statement based → via SQL proxy, such as “Pgpool-II”
    cf. Trigger based → Daemons run on both C/S, such as “Slony-I” & “Bucardo”
  - ✓ Log (i.e. binary block) based → PostgreSQL streaming replication
- BDR still has some constraints:
  - **i.** **“Primary key”** must be defined in every table. **“OID”** cannot be used.
  - ii. Data Definition Language (DDL) commands are not fully supported in BDR.
    - ➢ *e.g.* CREATE/DROP/ALTER DATABASE/ROLE/USER/GROUP/TABLESPACE/TYPE …
  - iii. BDR solves transaction conflicts using a simple **“last-update-wins”** strategy.
    - ➢ *Replication interval is set to* ***2 seconds*** *for better throughput.* → quasi- Real-time sync.

LHD data system

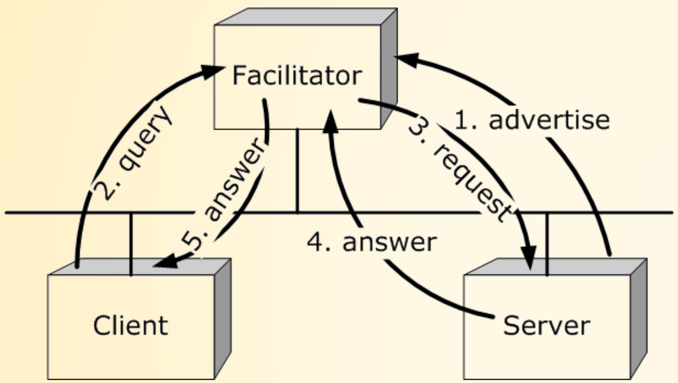
- Different from ITER UDA, LHD adopts “recommend” type of Facilitator model.
  → 2-step data access with different protocols
- LHD storage has 3 layers: ① SSD array, ② HDD raid cluster, ③ Blu-ray library.



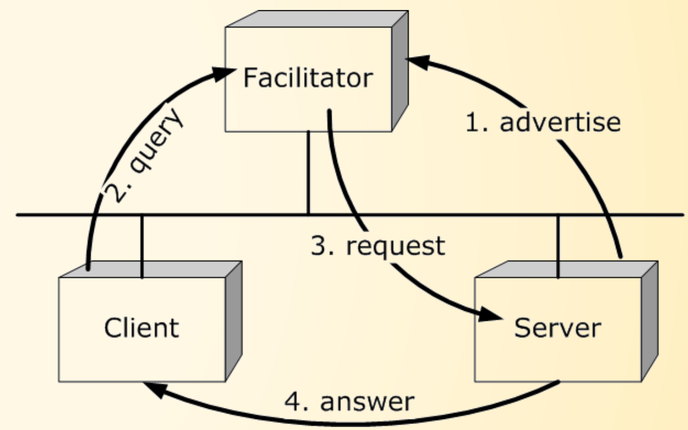
- Archiving data can be queued and also served in every stage.
- Real-time data streaming uses a simple C/S model and served independently from the discrete data service.

“Facilitator Model” for tripartite systems

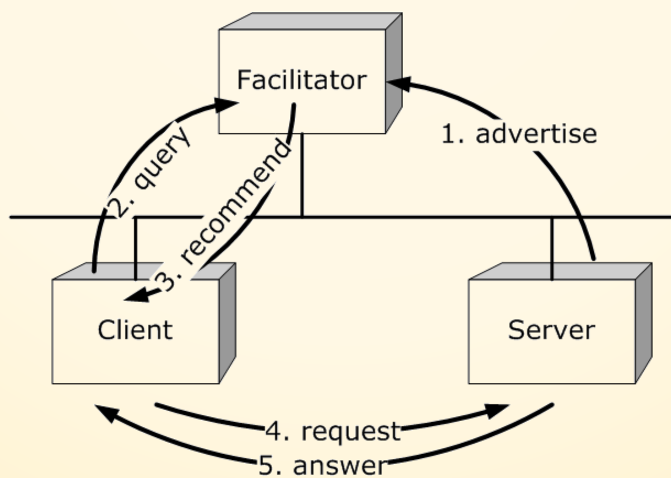
Broker type



Recruit type

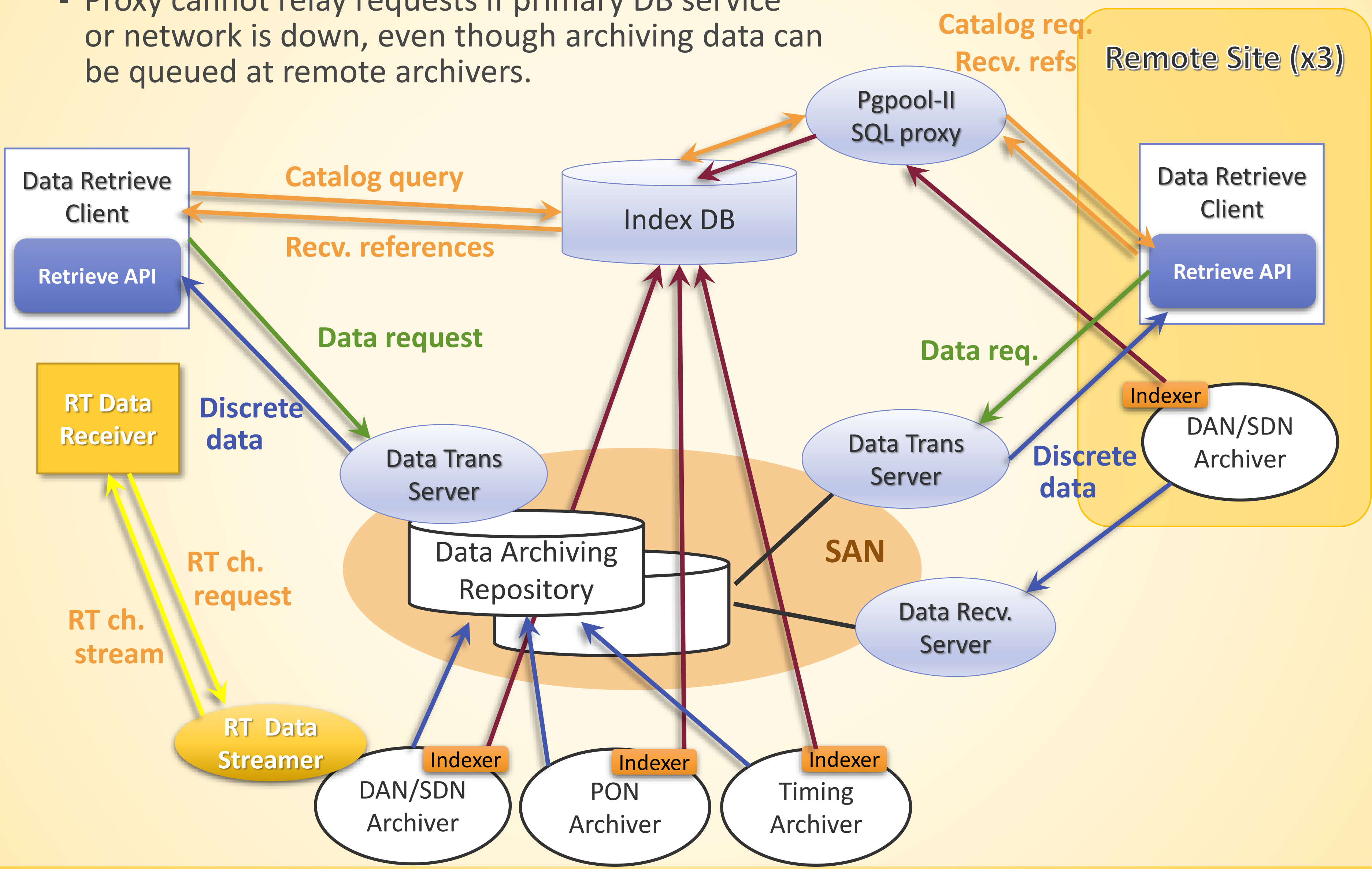


Recommend type



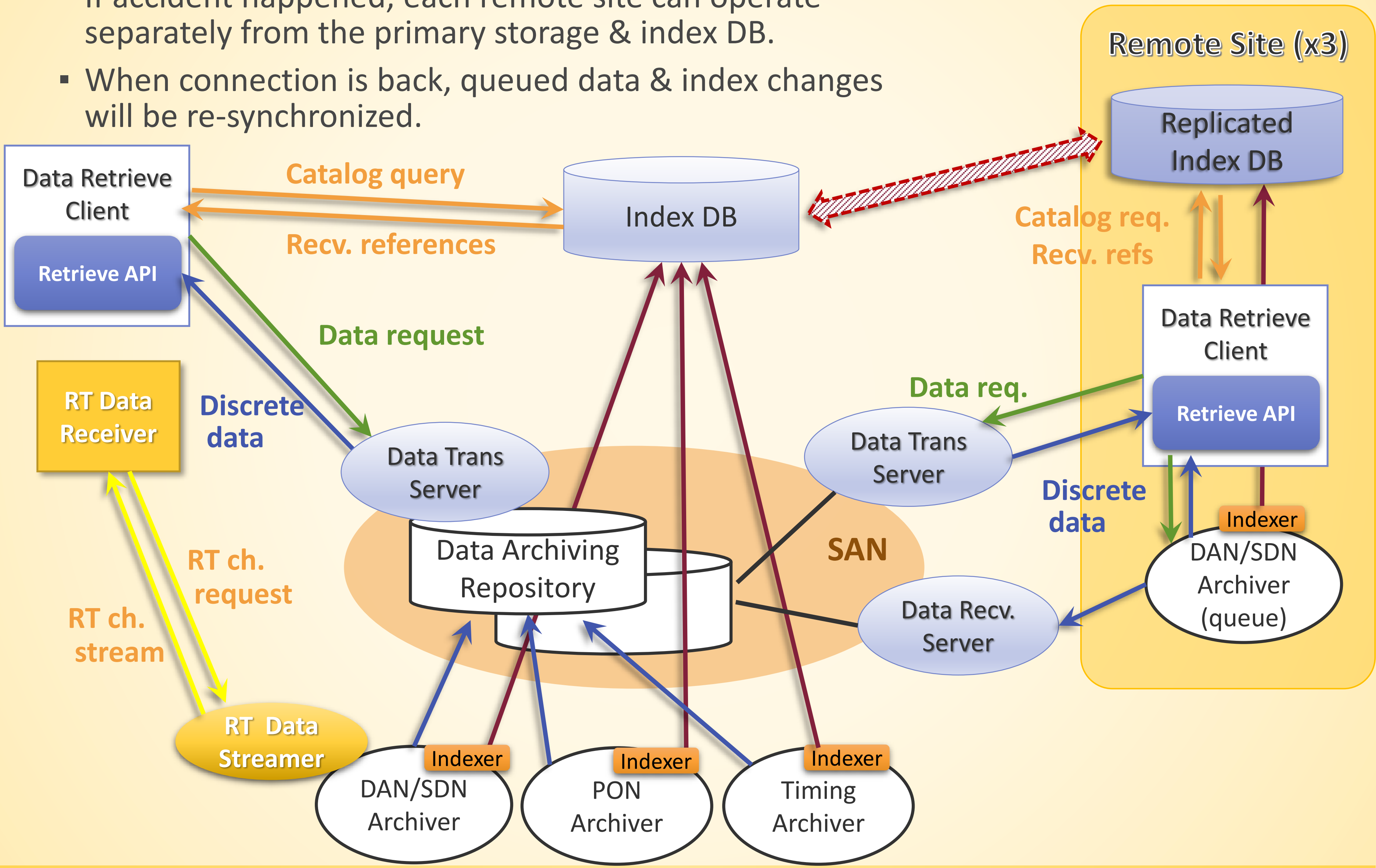
LHD data system having 3 remote sites (now)

- Proxy cannot relay requests if primary DB service or network is down, even though archiving data can be queued at remote archivers.



LHD data system having 3 remote sites (mod.)

- If accident happened, each remote site can operate separately from the primary storage & index DB.
- When connection is back, queued data & index changes will be re-synchronized.



Performances of Postgres BDR

- BDR throughputs have been investigated between NIFS, Toki and REC, Rokkasho.
  - **Round-trip time = 16.2 ms**, connected via 20 Gbps – 100 Gbps – 10 Gbps link
  - **Postgres BDR 9.4** servers := cpu: Xeon E5-2650 v4 2.2 GHz, 12c/24t, mem: 128 GB
    xfs: Samsung NVMe SSD 960 PRO 512GB

→ Replicating a single record may take **negligible small time (<< RTT)** on average for usual operations, excepting 2 second queuing.

Table Name (# of tables)	# of records	elapsed time	per record	note
Ex_Note (5)	144 772	243.6 s	16.8×10^{-4} s	--inserts
↑ BDR	144 772	23.1 s	1.60×10^{-4} s	(--copy)
↑ no BDR	144 772	0.978 s	6.76×10^{-6} s	(local)
Setup (167)	11 577 821	971.6 s	0.84×10^{-4} s	
↑ no BDR	11 577 821	62.15 s	5.37×10^{-6} s	(local)
Index (22)	207 911 053	35 654 s	1.72×10^{-4} s	-F c -j 3
↑ no BDR	237 544 798	581.5 s	2.45×10^{-6} s	-F c -j 3 (local)

Conclusions and Future works

- In order to put the replicated data repositories of practical use for massive data analyses, metadata Index DB should be also replicated for each repository site.
- Considering the compatibility of PostgreSQL, bi-directional replication extension Postgres BDR has been investigated and tested by using LHD Index data and FVL environment.
- BDR performance seems sufficient for usual data operations, excepting some global DDL commands.

In future works,

- A selection scheme for the most appropriate data repository site will be implemented soon in data retrieving client API and tested on LHD & FVL.
  - **List of possible Index DBs** can be stored and served by Index DB itself.
  - The best server can be found by practically measuring the network **round-trip time (RTT)** between C/S.
    - ✓ *e.g.* ICMP echo reply (ping) or TCP SYN+ACK response can be used.
    - ✓ DNS top domains or GeoIP resolvers provide answers with a very limited precision.

Thank you!