Distributed data storage for modern astroparticle physics experiments.

A. Kryukov*

SINP MSU

The 3-d Int. workshop
"Data Life Cycle-2019"
Irkutsk, 03-04.04.2019

Supported by RSF grants No.18-41-06003

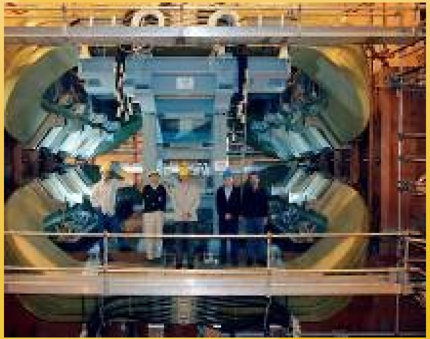
**E-mail: kryukov@theory.sinp.msu.ru*

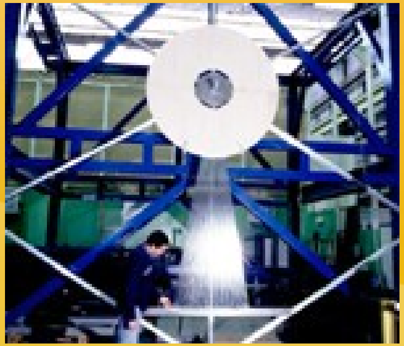
astroparticle.online

Examples of Large Experiments and Distributed Storages: WLCG

- The Worldwide LHC Computing Grid (WLCG)
  - It was designed by CERN to handle the prodigious volume of data produced by Large Hadron Collider (LHC) experiments in high-energy (elementary particle) physics
    - approximately 25 petabytes per year
  - an international collaborative project
  - **grid**-based computer network infrastructure incorporating over 170 computing/storage centers in 36 countries

[***CMS***](#) [*LHCb*](#) [*ATLAS*](#) [*ALICE*](#)



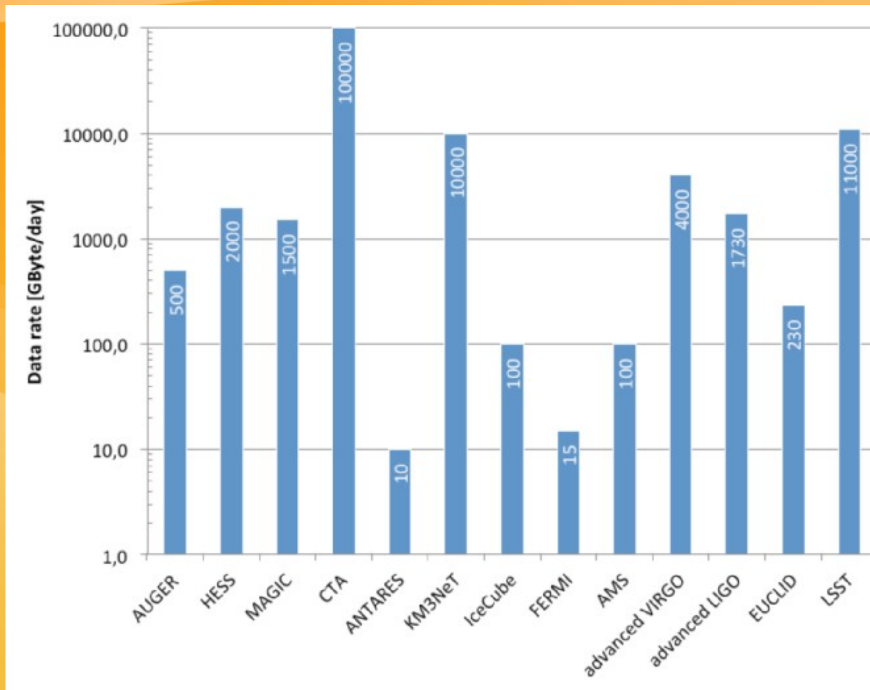


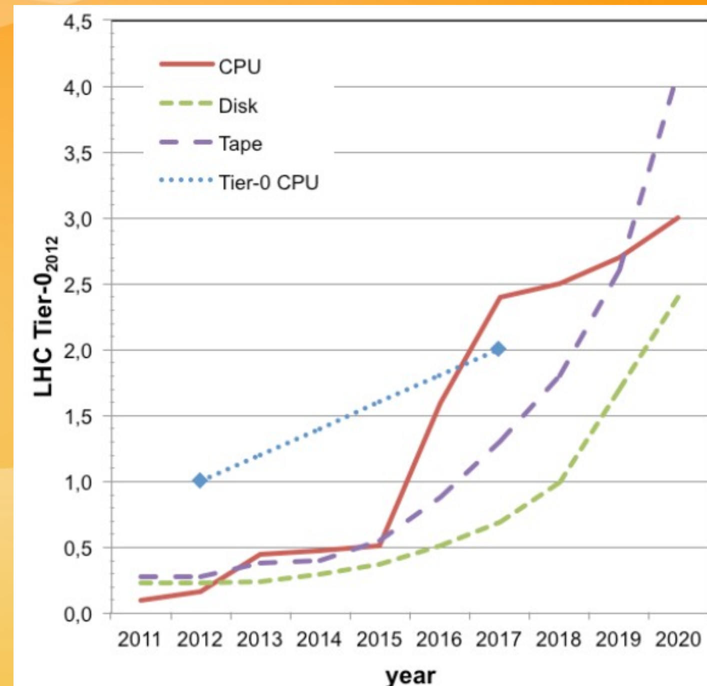
Multimessenger astronomy

astroparticle.online

Examples of Large Experiments and Distributed Storages: Astrophysics

001
101
.10
astroparticle.online





Berghöfer, T., et al. "Towards a model for computing in european astroparticle physics." ArXiv:1512.00988 (2015)

Data challenge in astronomy

astroparticle.online

- LSST imaging at this rate will generate about 15 terabytes (15 trillion bytes) of raw data per night and 30 petabytes over its 10-year survey life.
  - A petabyte is approximately the amount of data in 200,000 movie-length DVDs.
- Even after processing, that's still a 15 PB (15,000 TB) store.

Sky Survey Projects	Data Volume
DPOSS (The Palomar Digital Sky Survey)	3 TB
2MASS (The Two Micron All-Sky Survey)	10 TB
GBT (Green Bank Telescope)	20 PB
GALEX (The Galaxy Evolution Explorer)	30 TB
SDSS (The Sloan Digital Sky Survey)	40 TB
SkyMapper Southern Sky Survey	500 TB
PanSTARRS (The Panoramic Survey Telescope and Rapid Response System)	~ 40 PB expected
LSST (The Large Synoptic Survey Telescope)	~ 200 PB expected
SKA (The Square Kilometer Array)	~ 4.6 EB expected

Architecture of DS for megascience experiments

astroparticle.online

Analysis and Data Center in Astroparticle Physics

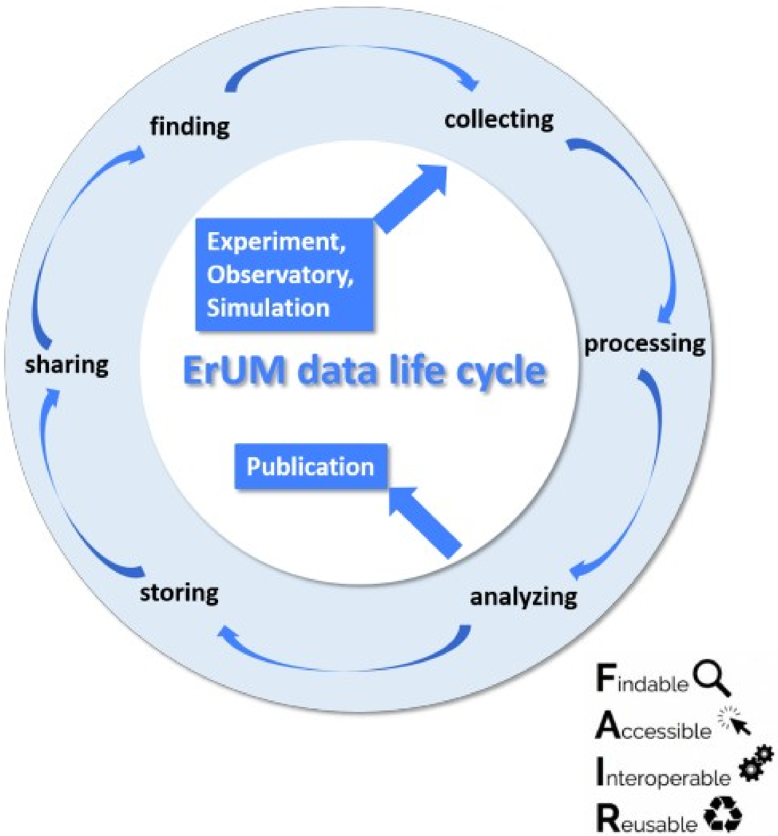
Data availability | Analysis | Simulations & Methods development | Open access | Education in Data Science | Data archive

Partly realized in individual experiments

- **Data availability:**
  All researchers of the individual experiments or facilities require quick and easy access to the relevant data.
- **Analysis:**
  Fast access to the generally distributed data from measurements and simulations is required. Corresponding computing capacities should also be available.
- **Simulations and methods development:**
  The researchers need an environment for the production of relevant simulations and the development of new methods (machine learning).
- **Open access:**
  More and more it is necessary to make the scientific data available not only to the internal research community, but also to the interested public: public data for public money!
- **Education in data science:**
  Not only data analysis itself, but also the efficient use of central data and computing infrastructures requires special training.
- **Data archive:**
  The valuable scientific data and metadata must be preserved and remain interpretable for later use (data preservation).

Research Data Management

- **Where possible, common standards should be established to foster interoperability**
- **Importance of "data stewards" to manage the data life cycle and to act as a curator for metadata**

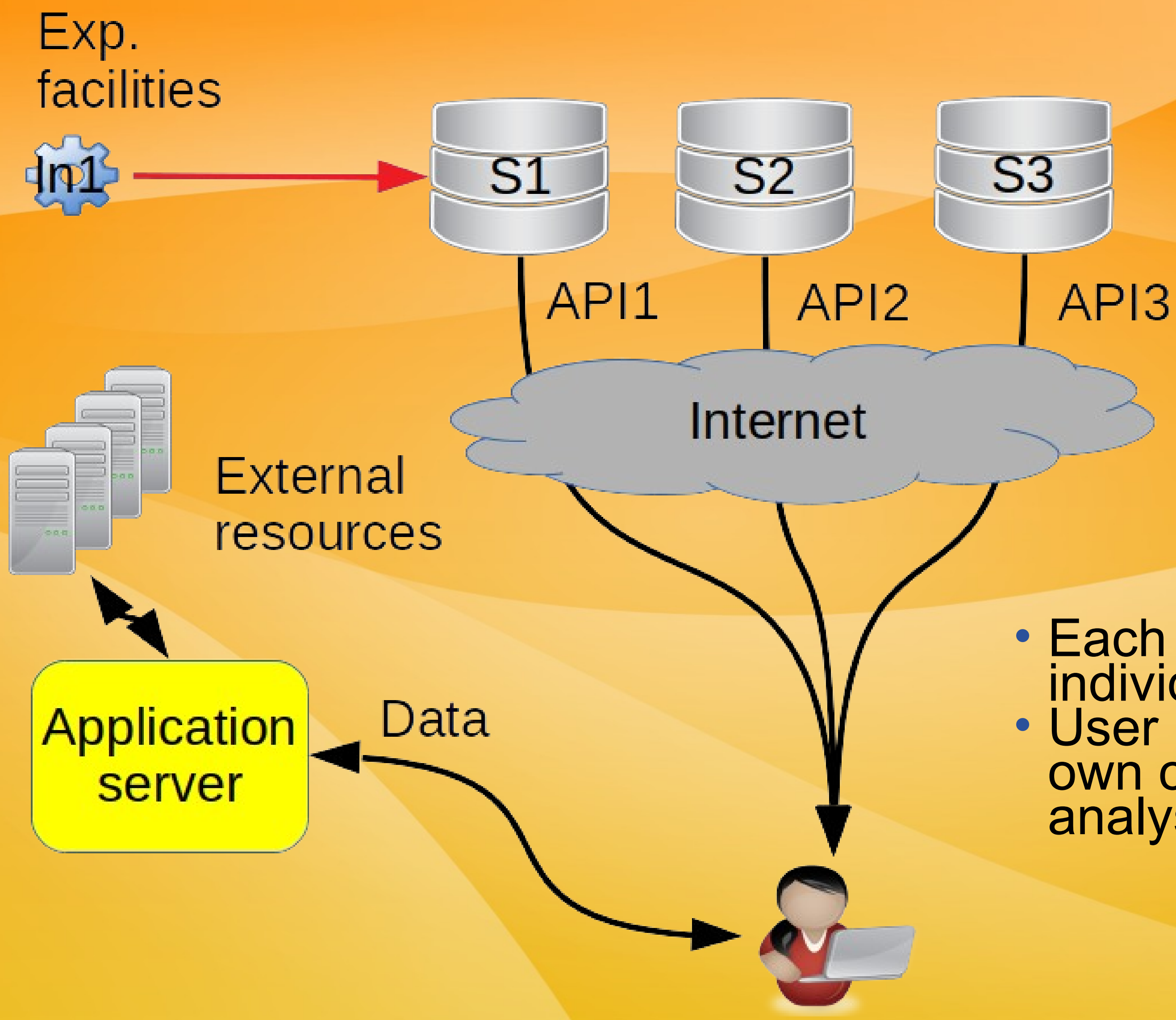


Motivation

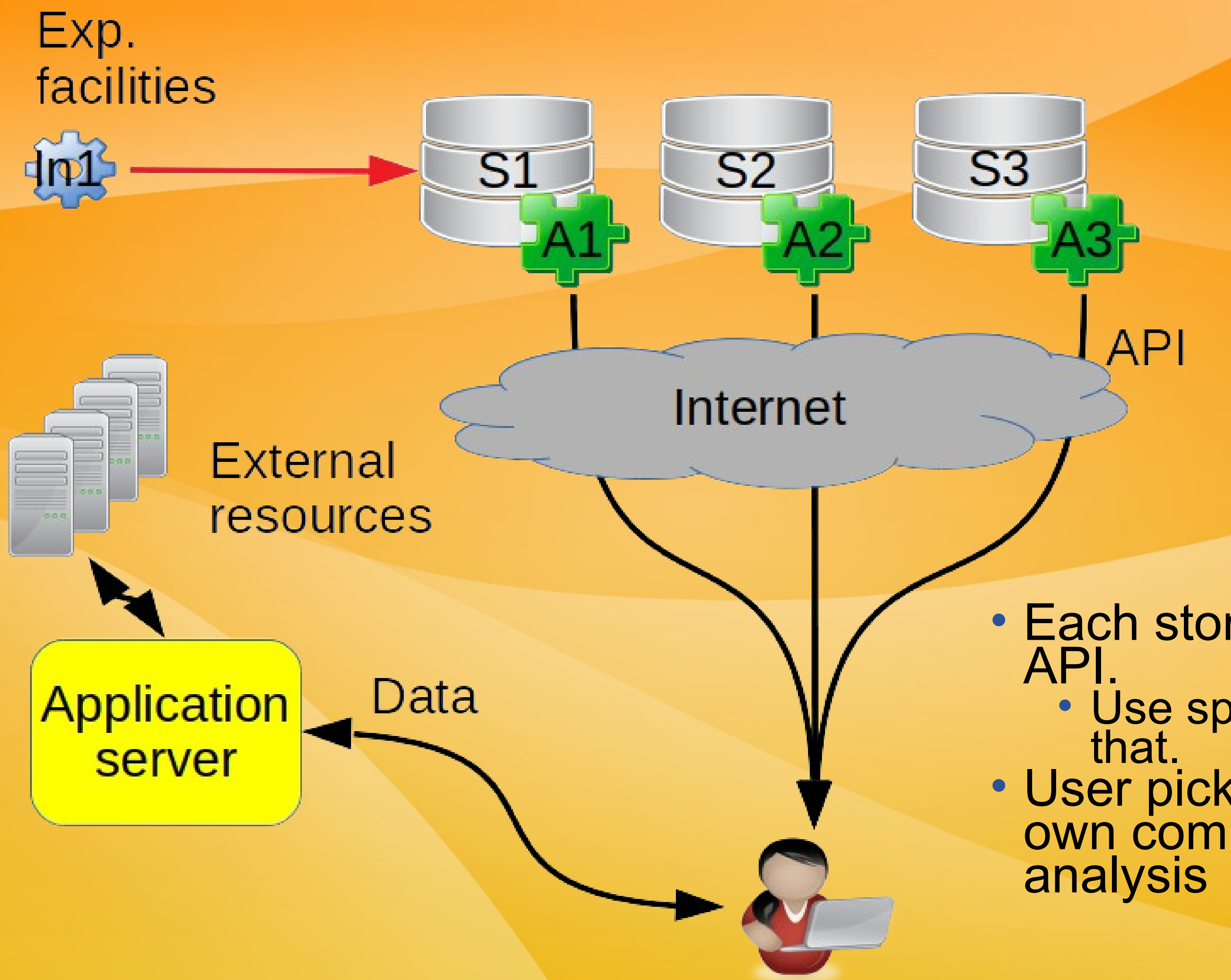
- Modern installations generate terabytes and petabytes of data. The collaboration brings together hundreds and thousands of researchers from many organizations. Thus, it is necessary to organize access for scientists who use these repositories to perform data analysis.
- Most collaboration store data as a collection of files.
  - Special case: DB-oriented storage (KASCADE).
- They have a long history and established practice of working with data.
  - No change to existing site infrastructure, only add-ons
- Open Science is a modern trend in the physics.

The main ideas that are embedded in the architecture

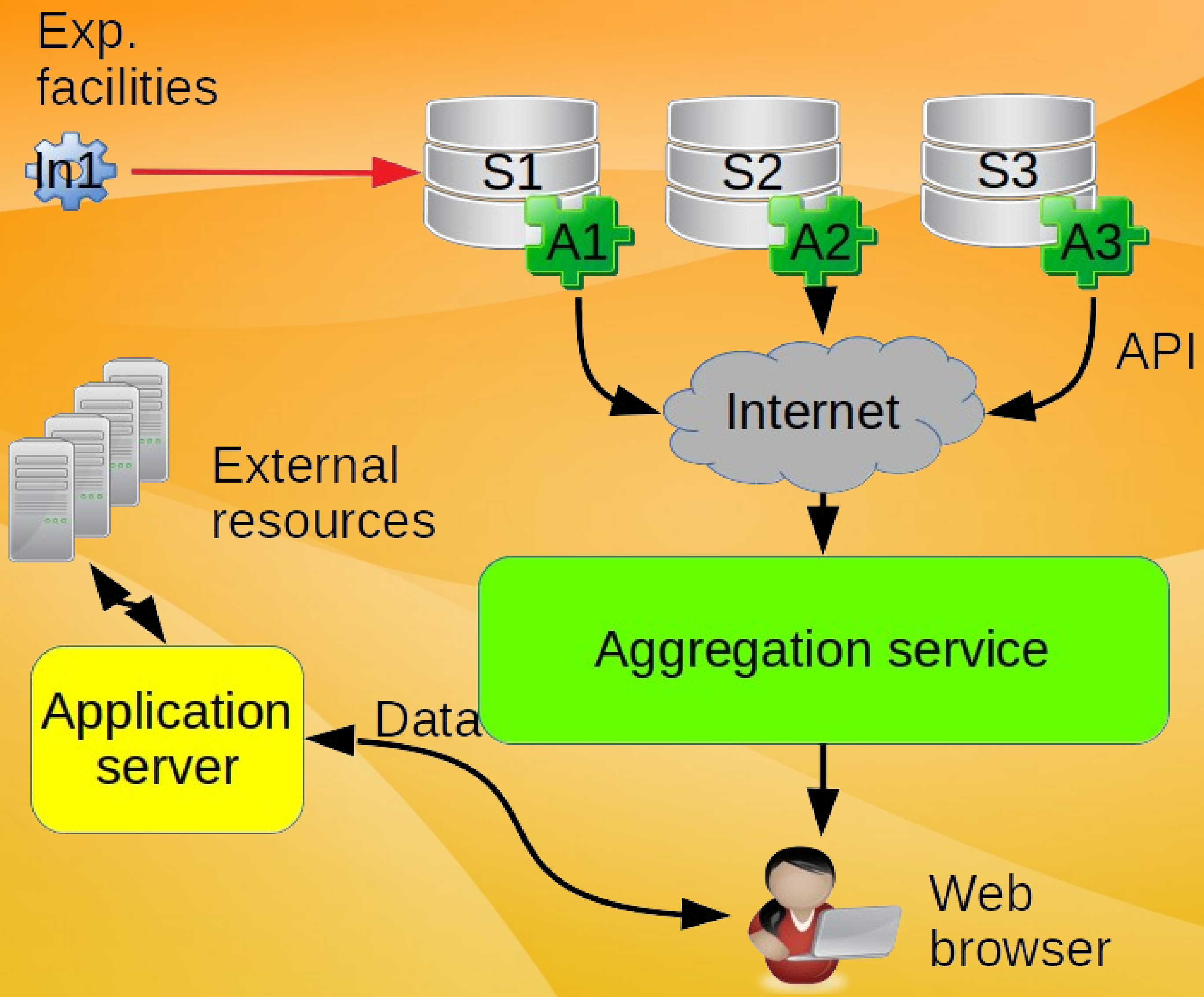
- Read-only oriented distributed storage (DS).
  - Remote access to data as local file systems
  - On-demand data transfer by requests only
- No intervention into local storage, special adapters are used to access data.
- User requests are processed on a dedicated server based on metadata only.
- Metadata is extracted from primary and/or secondary data in semi-automatic mode.
  - Binary format description language is used for serialize/deserialize binary data.



- Each storage has individual API
- User pick up data to the own computer and make analysis



- Each storage has unified API.
  - Use special adapters for that.
- User pick up data to the own computer and make analysis



Exp.
facilities
In1
S1
S2
S3
A1
A2
A3
API
Internet
External
resources
Aggregation service
Application
server
Data
Web
browser

CERN VM-FS as an adapter

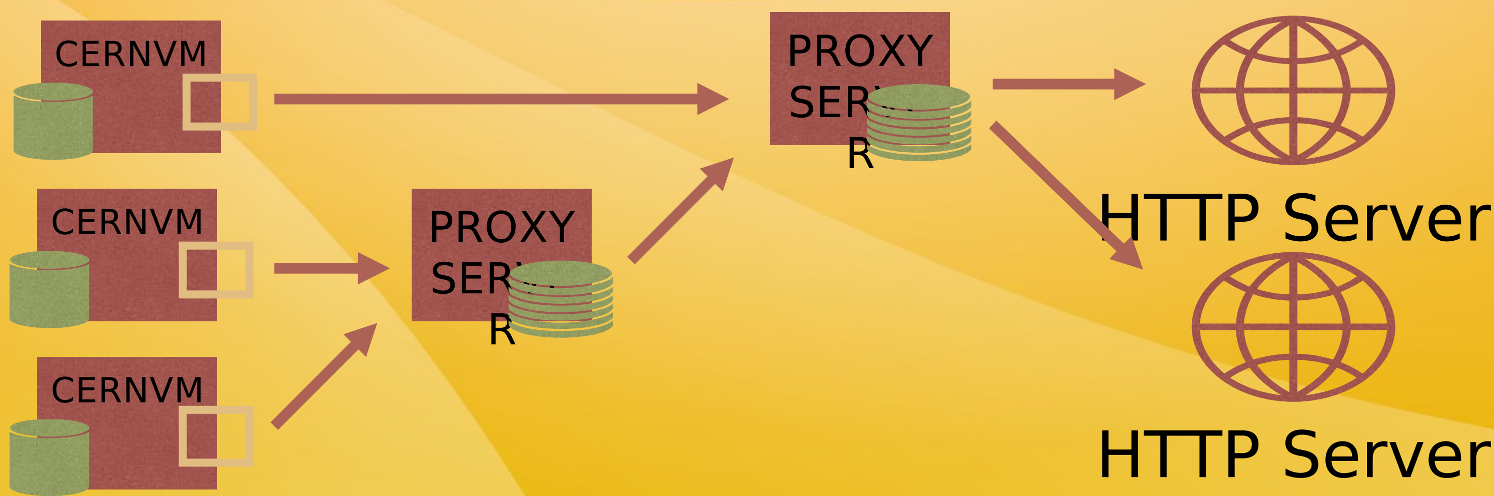
- Data are left untouched in their own file system
- CernVM-FS indexes the data and changes, stores only the metadata (indices, checksums, locations, etc.) and data tree
- CernVM-FS uses HTTP as the data transfer protocol, so there’s no firewall problem
- Data transfer starts only on actual reads
- Multilevel cache-proxy servers

CERNVM-FS

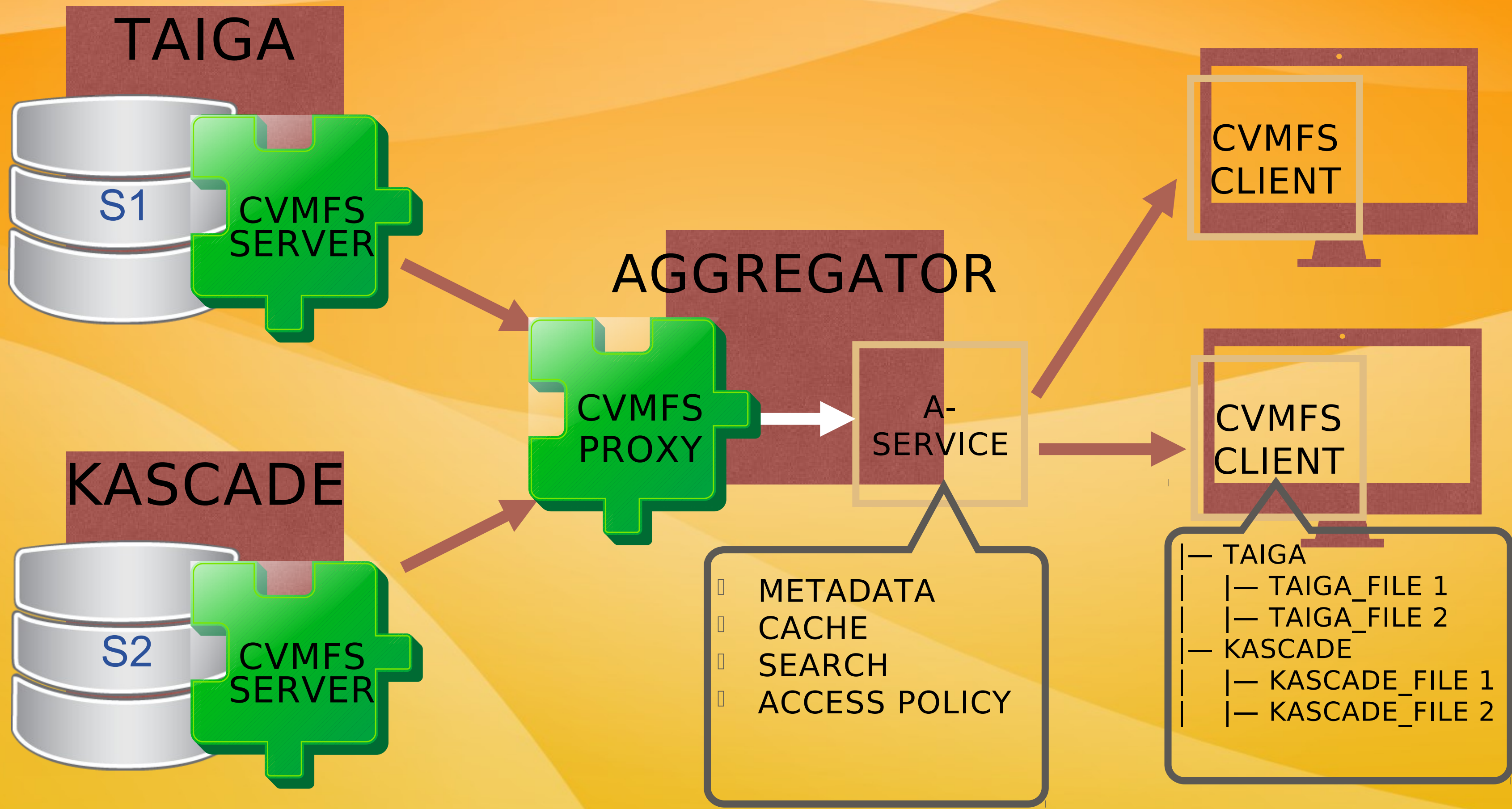
astroparticle.online

- DATA UPDATE

- DATA DISTRIBUTION

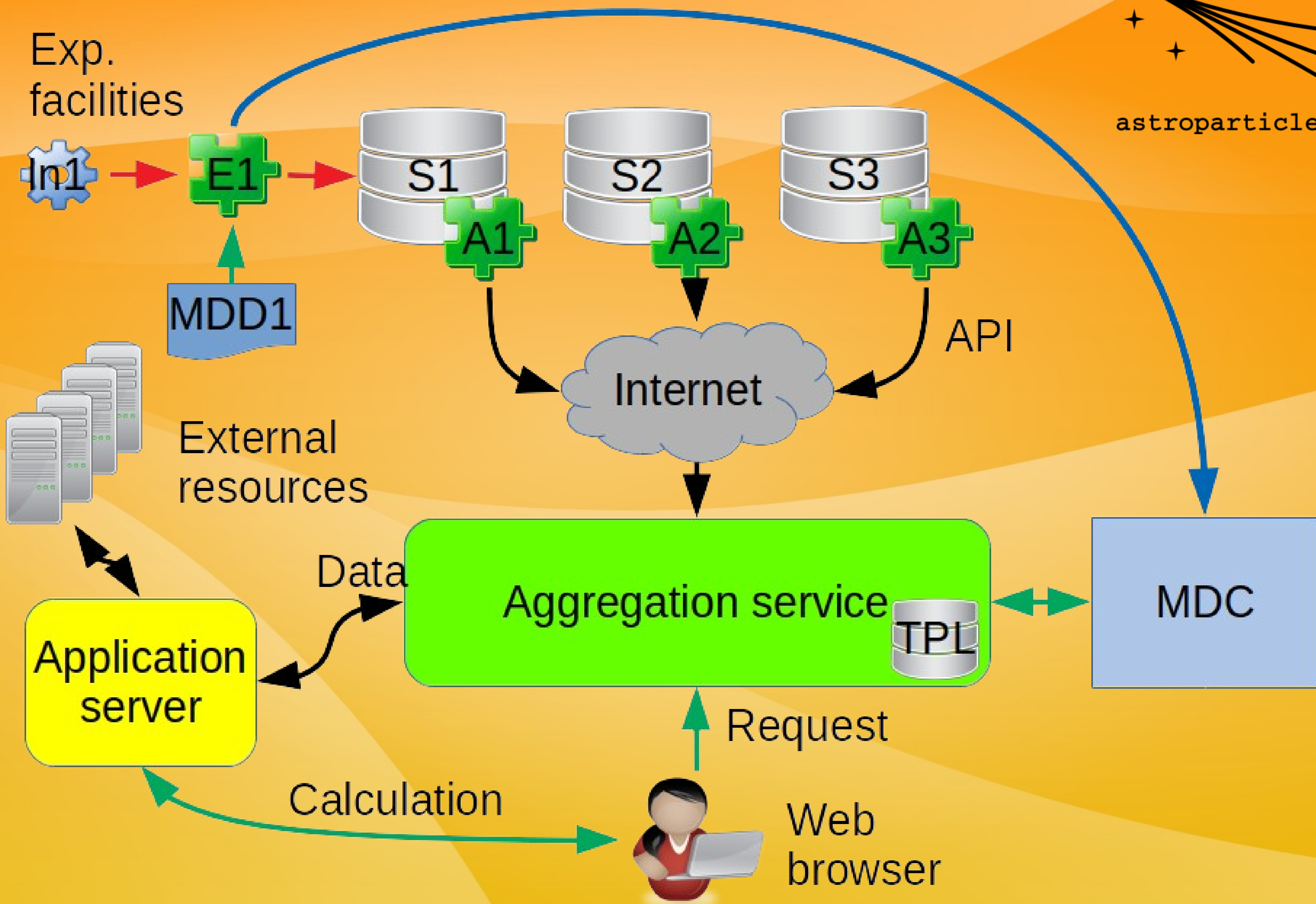


CERNVM-FS

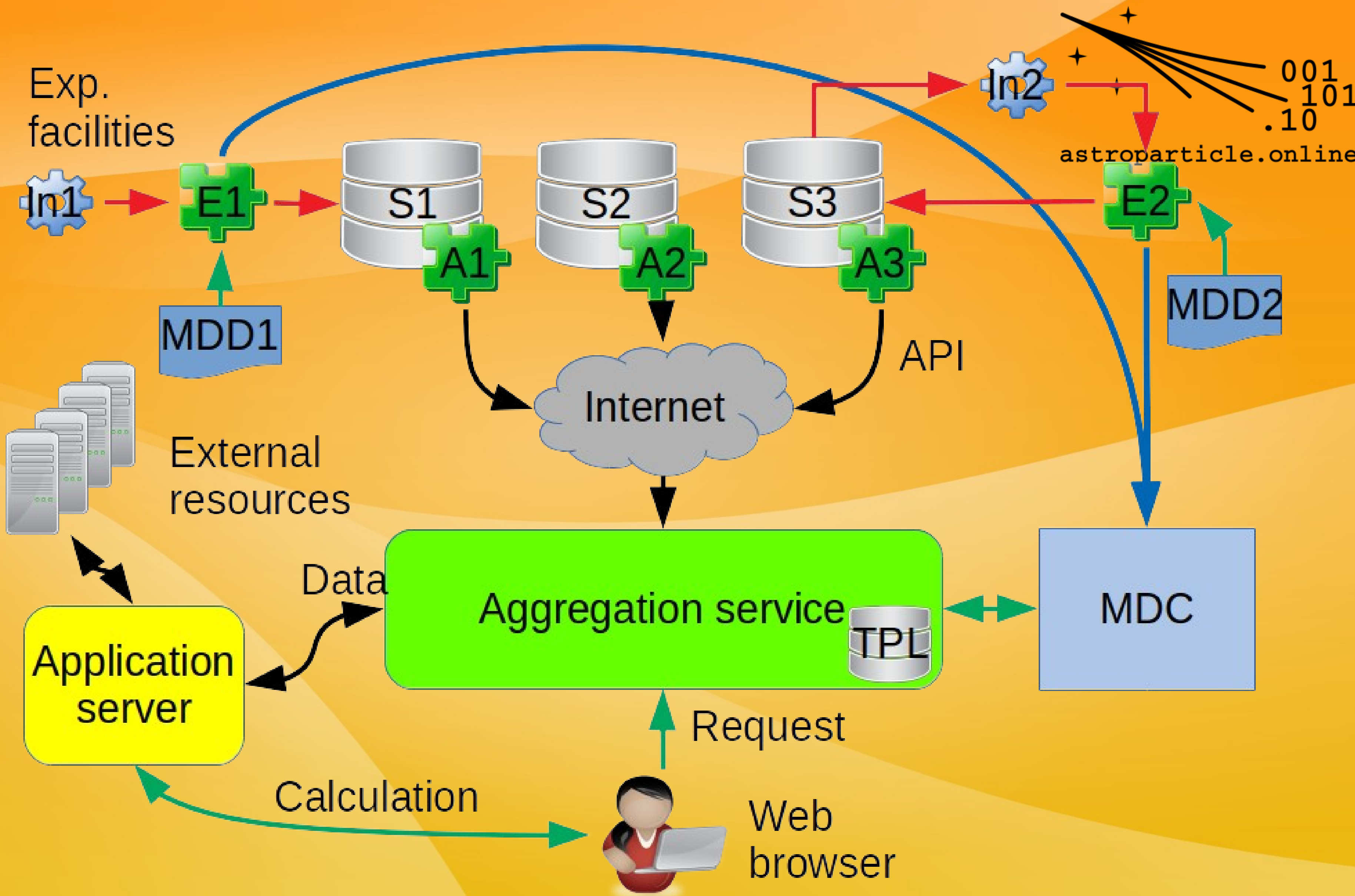

TAIGA
S1
CVMFS SERVER
KASCADE
S2
CVMFS SERVER
AGGREGATOR
CVMFS PROXY
A-SERVICE
METADATA
CACHE
SEARCH
ACCESS POLICY
CVMFS CLIENT
CVMFS CLIENT
|— TAIGA
| |— TAIGA_FILE 1
| |— TAIGA_FILE 2
|— KASCADE
| |— KASCADE_FILE 1
| |— KASCADE_FILE 2

Information system

- Any user request is processed on Metadata catalogue (MDC) service.
  - No direct request to the local storage.
- We use special programs – called extractors to extract metadata.
- Two level metadata:
  - File level MD (experiment, detector, date, session,…)
  - Event level MD (energy, type of primary particle, …)
    - This MD is usually the result of raw data processing.



astroparticle.online


Exp.
facilities
In1
E1
S1
S2
S3
A1
A2
A3
MDD1
In2
001
101
.10
astroparticle.online
E2
MDD2
API
Internet
External
resources
Data
Aggregation service
TPL
MDC
Application
server
Request
Calculation
Web
browser

Metadata extractors

- Extract meta data from primary and/or secondary data
- The list of extracted MD should cover all allowed user requests.
- The answer of MDC is a list of URL(URI) where necessary data locate.

Kaitai Struct

- Declarative language for describing binary formats.
- Allows:
  - Describe the format in YAML
  - Check using visualization tools (ksv)
  - Compile to library in target language (ksc)
  - Use received API
- License
  - Compiler - GPLv3 +
  - Library for reading files - MIT or Apache v2

Kaitai Target languages

{.js} JavaScript

C++

python™

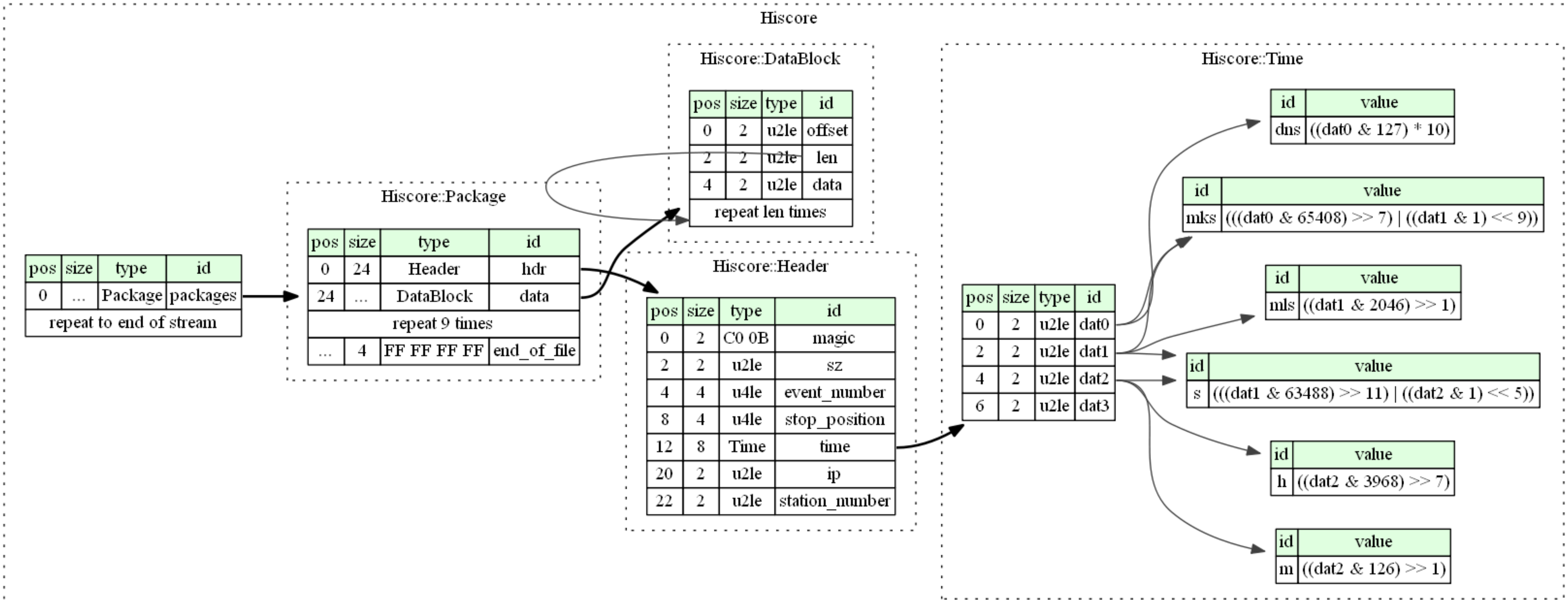
Java™ ORACLE

Go

Lua

Etc

MD description. HiScore example



HiSCORE format specification expressed in Kaitai Struct

Part I

```yaml
meta:
  id: hiscore
  title: HiSCORE data
  license: Unlicensed
seq:
  - id: packages
    type: package
    repeat: eos
types:
  package:
    seq:
    - id: hdr
      type: header
    - id: data
      type: data_block
      repeat: expr
      repeat-expr: 9
    - id: end_of_package
      contents:
      [0xFF, 0xFF, 0xFF, 0xFF]
```

Part II

```yaml
header:
  seq:
  - id: magic
    contents: [0xC0, 0x0B]
  - id: sz
    type: u2le
  - id: event_number
    type: u4le
  - id: reserved
    type: u4le
  - id: time
    type: time
  - id: ip
    type: u2le
  - id: station_number
      type: u2le
data_block:
  seq:
  - id: offset
    type: u2le
  - id: len
    type: u2le
  - id: data
    type: u2le
    repeat: expr
    repeat-expr: len
```

Part III

```yaml
time:
  seq:
  - id: dat0
    type: u2le
  - id: dat1
    type: u2le
  - id: dat2
    type: u2le
  - id: dat3
    type: u2le
  instances:
    dns:
    value: '(dat0 & 0x7f) * 10'
    mks:
    value: '((dat0 & 0xff80) >> 7) | (dat1 & 1) << 9'
    mls:
    value: '(dat1 & 0x7fe) >> 1'
    s:
    value: '((dat1 & 0xf800) >> 11) | ((dat2 & 1) << 5)'
    m:
    value: '(dat2 & 0x7e) >> 1'
    h:
    value: '(dat2 & 0xf80) >> 7'
```

HiScore Kaitai auto generates program

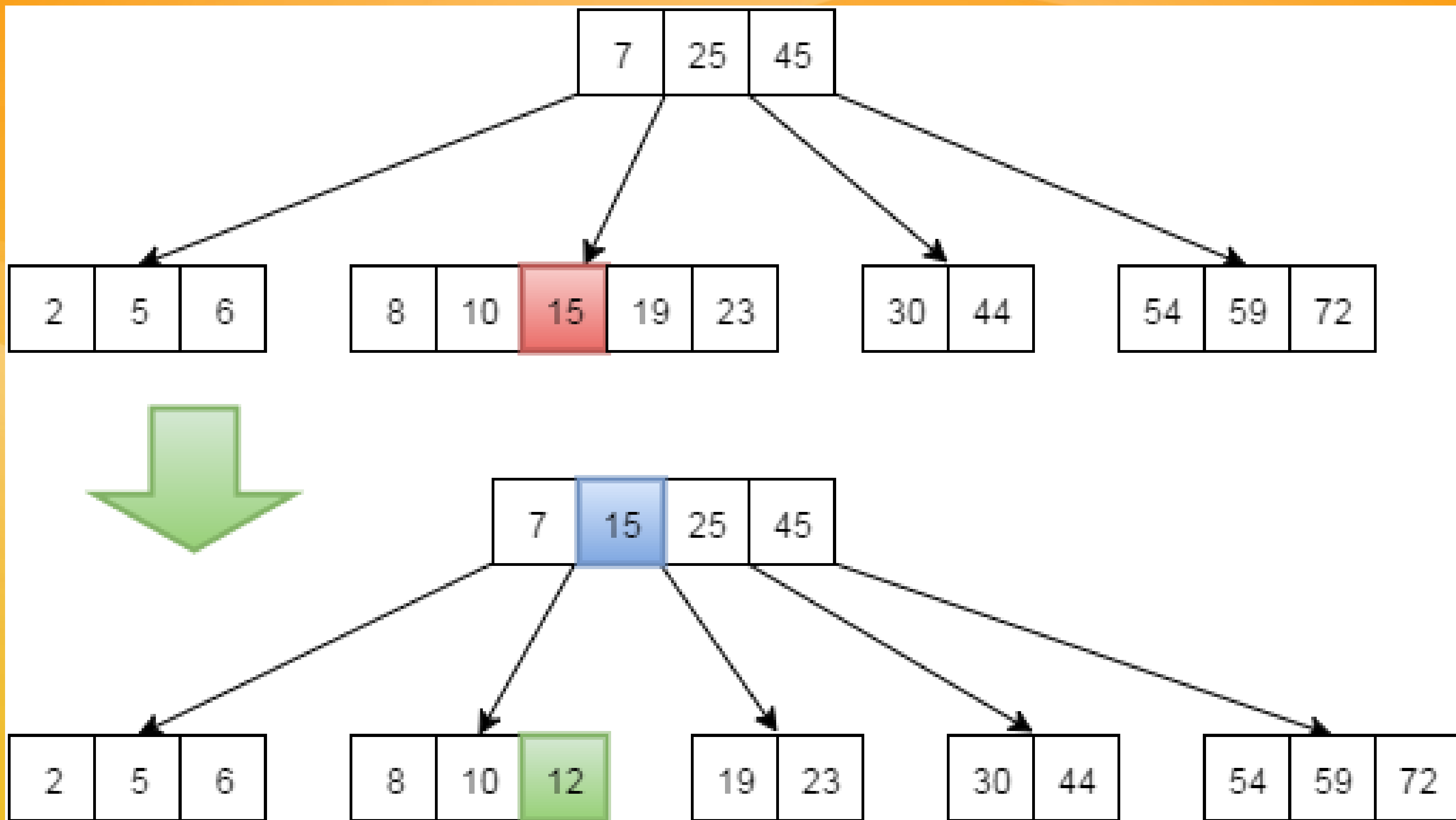
astroparticle.online

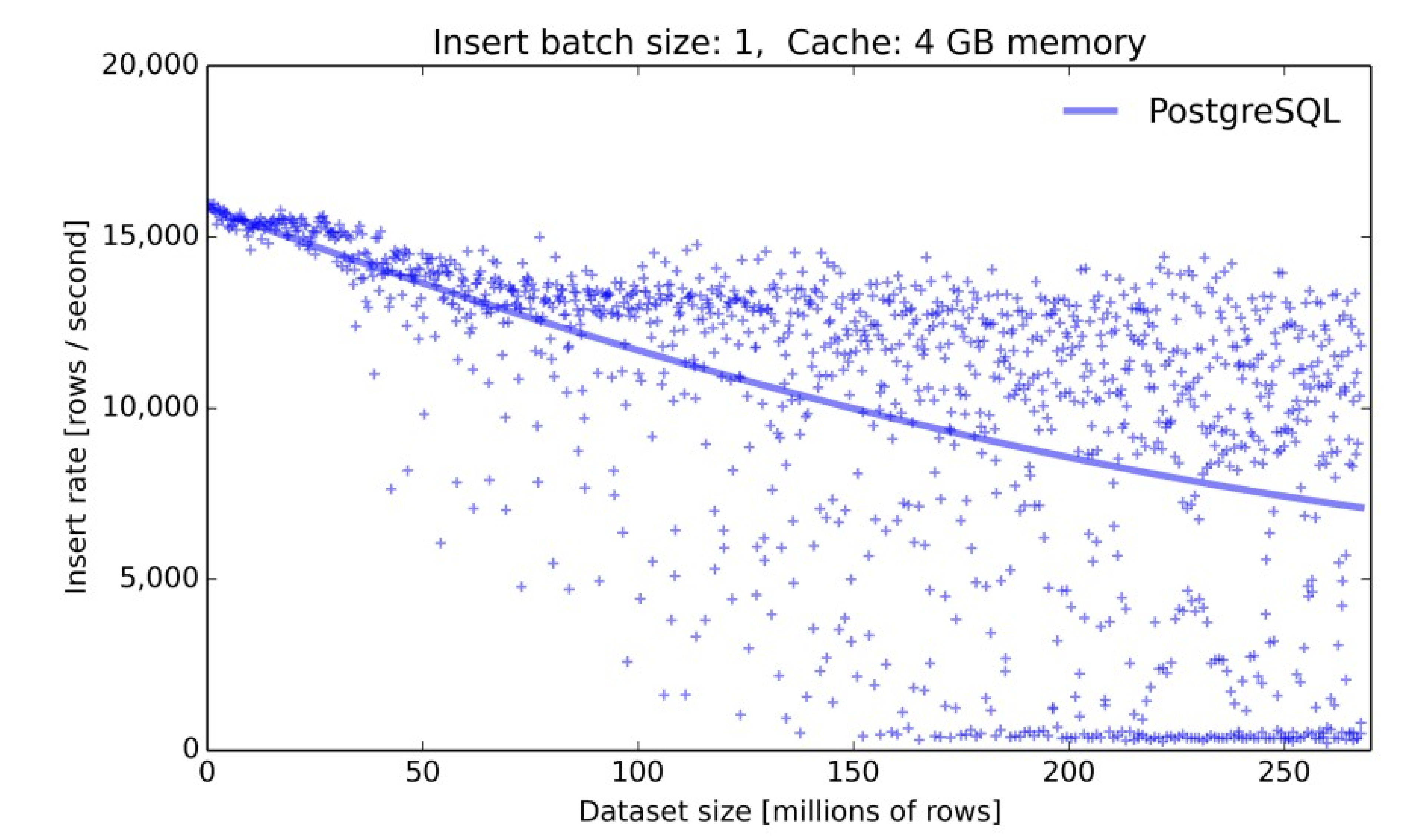
```java
public static void main(String[] args) throws IOException {
        Hiscore hiscore = Hiscore.fromFile(FILE_NAME);
        int pckgNumber = 0;
        for (Hiscore.Package pckg : hiscore.packages()) {
                System.out.println("Package: " + pckgNumber);
                System.out.println(String.format("Magic: %02X %02X", pckg.hdr().magic()[0], pckg.hdr().magic()[1]));
                System.out.println("Size: " + pckg.hdr().sz());
                System.out.println("Event number: " + pckg.hdr().eventNumber());
                System.out.println("Stop position: " + pckg.hdr().stopPosition());
                System.out.println(String.format("H %d: M %d: S %d: DNS %d: MKS %d: MLS %d",
                        pckg.hdr().time().h(), pckg.hdr().time().m(),
                        pckg.hdr().time().s(), pckg.hdr().time().dns(),
                        pckg.hdr().time().mks(), pckg.hdr().time().mls()));
                System.out.println("IP: " + pckg.hdr().ip());
                System.out.println("Station number: " + pckg.hdr().stationNumber());
                int size = pckg.data().size();
                int dataBlockNumber = 0;
                for (Hiscore.DataBlock data: pckg.data()) {
                        System.out.println("Data block: " + dataBlockNumber);
                        System.out.println("Offset: " + data.offset());
                        System.out.println("Length: " + data.len());
                        int n = data.len() > DATA_COUNT_TO_VIEW ? DATA_COUNT_TO_VIEW : size;
                        System.out.print("[");
                        for (int i = 0; i < n; i++) {
                                System.out.print(data.data().get(i) + ", ");
                        }
                        System.out.print("...]");
                        System.out.println();
                        ++dataBlockNumber;
                }
                ++pckgNumber;
}
```

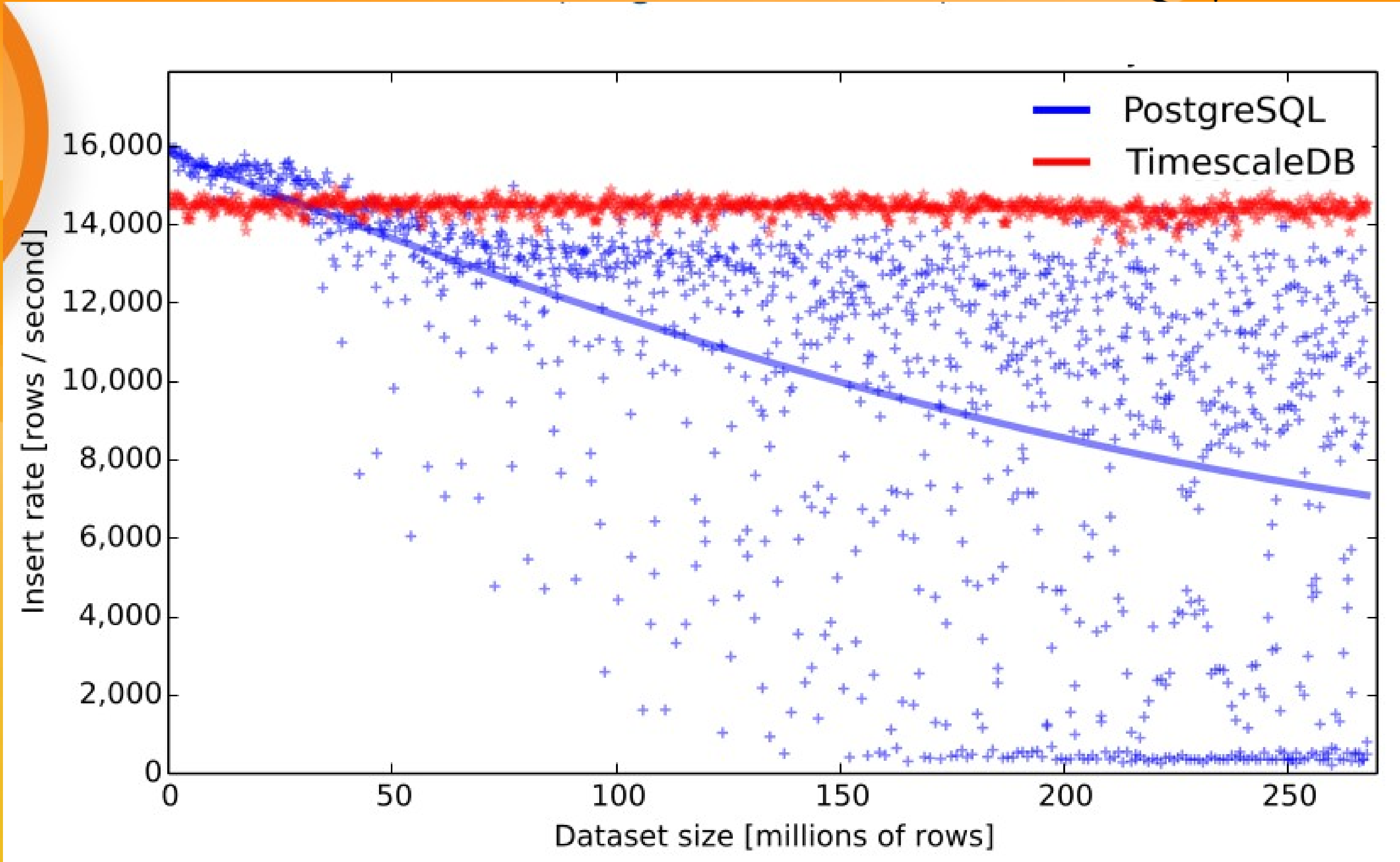
DB for information system

- Relation DB (MySQL, PostgreSQL, …)
  - Degradation of speed.
  - Reshuffle of indexes under massive insert operations.
- Time series DB (TimeScale DB)

Insertion of element into B-tree




Insert batch size: 1, Cache: 4 GB memory
PostgreSQL
Insert rate [rows / second]
20,000
15,000
10,000
5,000
0
0
50
100
150
200
250
Dataset size [millions of rows]


PostgreSQL
TimescaleDB
Insert rate [rows / second]
16,000
14,000
12,000
10,000
8,000
6,000
4,000
2,000
0
0
50
100
150
200
250
Dataset size [millions of rows]

File level request processing

- MDC returns a list of URLs where the required data is located.
- The aggregation service re-exports to the user only those files that are in the list, obtained from MDC.

Event level request processing

- MDC returns a list of URLs where the required data is located.
- The aggregation service scans these files and extracts those events that satisfy the user's request.
- As a result, a new collection is formed, which contains only the necessary events.
- This collection is exported to the user.

Conclusions

- Modern information technologies can provide scientists with convenient access to large data distributed throughout the world.
- Distributed storage provides analysis of instant messengers and intelligent management of data access rights.
- A custom data request can contain both file level filters and more detailed filters, such as events.


Any questions?