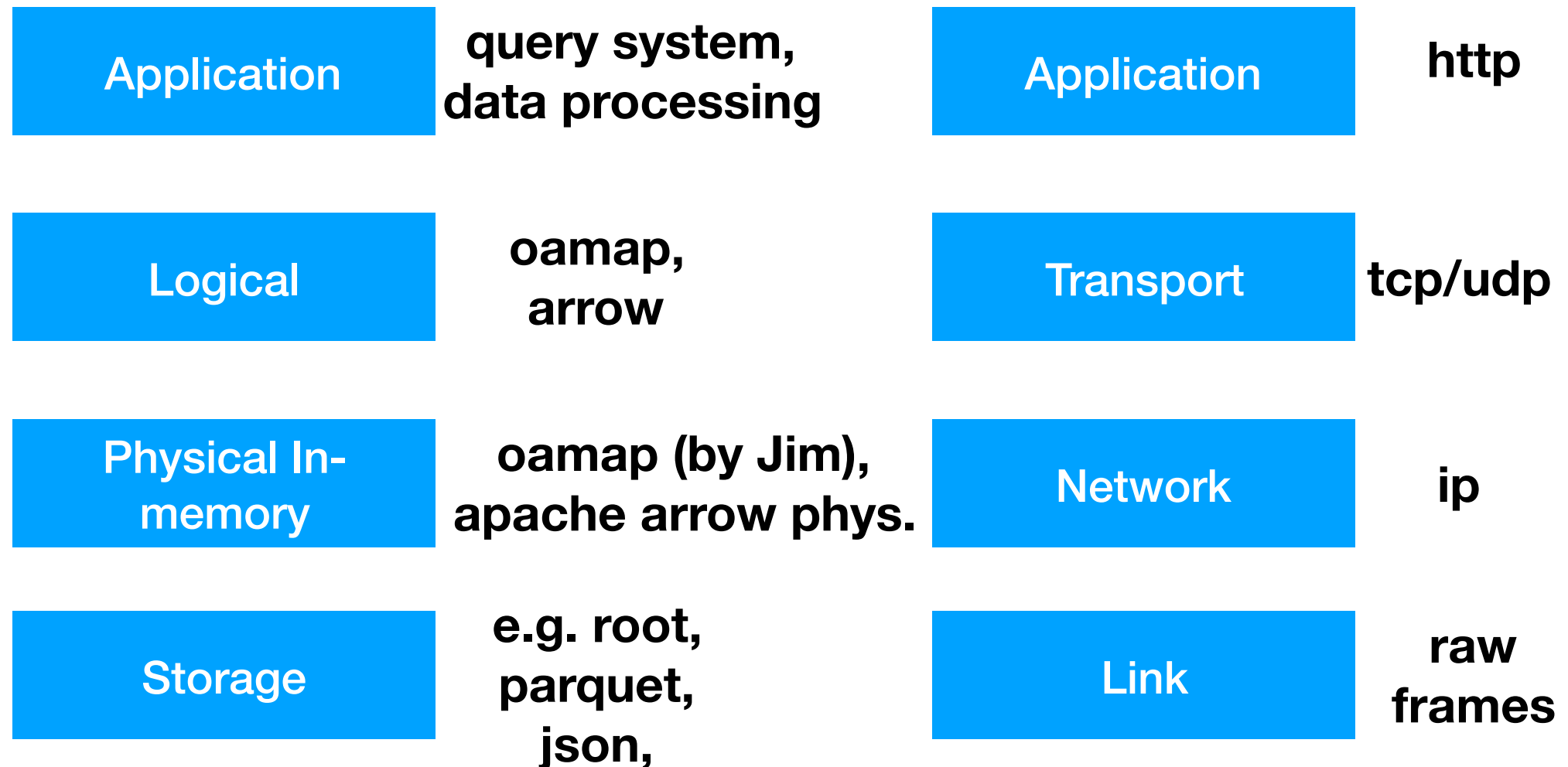


DMS - Data Model Stack

DMS

- Data Model Stack - a layered interface specification of the data model
- Specify only interfaces between layers, not implementation
- Inspired by OSI, TCP/IP
- Abstracts the data storage, physical, logical representations.

Stack



...

**Type System (Storage) \leq Type System (Physical)
 \leq TS (Logical) \leq TS (Application)**

**Storage Type System = {fixed bit width primitives AND
arrays of fixed bit width prims}**

Where is root in here???

- ROOT collapses all the layers into just 1! ROOT serializes as 1-to-1 mapping
- very nice from programming point of view
- bad for heterogeneity, EDM in general is bad for heterogeneity.
- can that scale??? what else can be done to this model of 1-to-1 serialization???
- SKA, for instance, have matrices (a single matrix object so to speak...) which do not fit into memory of a single machine.

Where Apache Arrow in this

- Working on using ROOT as storage and Arrow as in-memory representation right now.
 - analogous to parquet-cpp
 - Need to understand arrow first 100%
- Physical and Logical Apache Arrow Specification is just a particular implementation. Buffers could be encoded (implemented) differently...
- The type system does not include pointers for instance...
- It's a new project
- “As minor breaking API change in C++, we have made the `RecordBatch` and `Table` APIs “virtual” or abstract interfaces, to enable different implementations of a record batch or table which conform to the standard interface. This will help enable features like lazy IO or column loading.”