Data Access via TEvent/TStore

TEvent: combines basic functionality of StoreGate, TSelector, OutputStream
TStore: transient-only data store

Not sure why two very similar APIs

- Objects can be shared(copied?) between the two, allowing to save TStore objects

How about merging them?

Private Stores

Use Cases:

Train Carriages

- One store per carriage

Systematics Variations

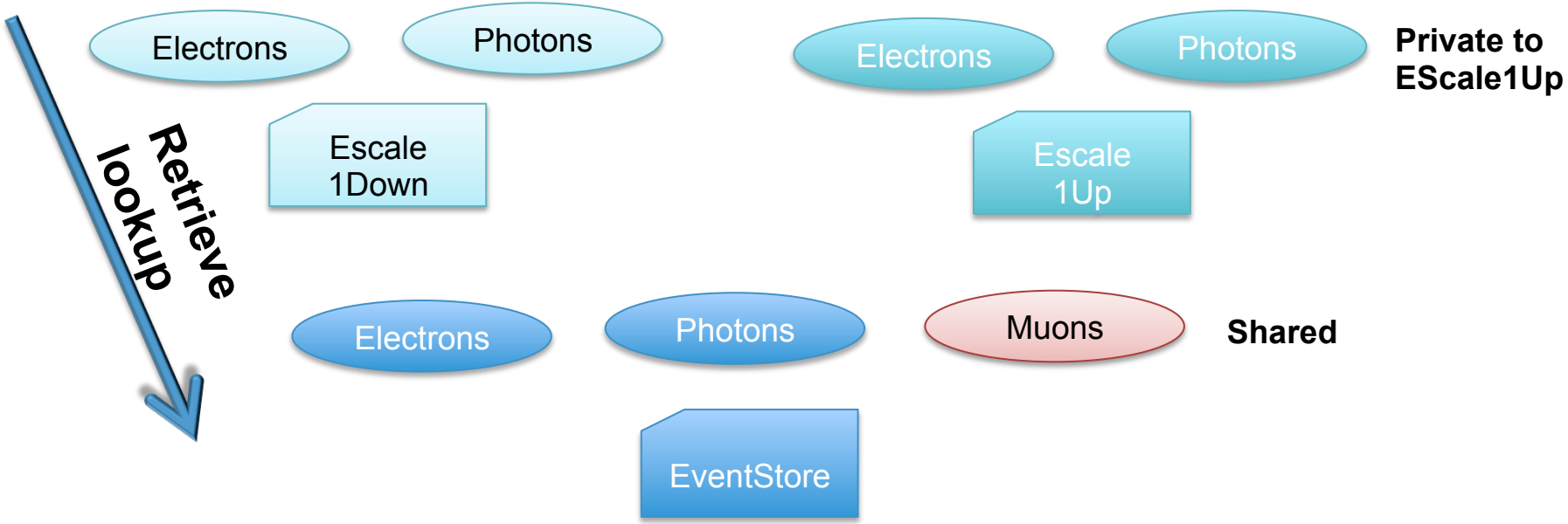
- Multiple object versions

(HLT ROIs and Trigger Steps)

- *Event Views*

- Insulate execution of Algorithms/Tools
- Slice event context
- Allow both shared and context-specific data objects
- Single context switch

Common Theme: Inverted Lookup



Design Alternatives

SG Folders:

- like TDS directories (or Trigger Elements)
- Reverse “find”: 1st current dir, then up

Context-specific View Stores:

- Shallow copy event store into context store
- Record context-specific data objects

VarHandles in Dual-Use Tools?

Declared and set as an AlgTool Property

In .h

```
SG::RVar<EventInfo> m_r_evtInfo;
SG::WVar<int> m_w_int;
```

In .cxx

```
declareProperty("EvtInfoHandle",
 m_r_evtInfo,
"An EventInfo read handle");
```

Pointer-like syntax

```
m_w_int = new int(
 m_r_evtInfo->event_number())
```

Prototype in athena

- a cross between `DataLink` and `ToolHandle`
- replace record/retrieve
- Interface not as rich as StoreGateSvc

Soon to use in athena

- how to avoid split with dual-use tools

Status of VarHandles

Goal is to have them “tutorial-ready” for Sep
(already presented in July LBL tutorial)
Focus on interface, some tuning needed

- Find better name :-)
- How to connect to a non-default store
- What is a valid handle? How to do

```
if (!contains(key)) record(dobj,key)
```

- Optimize implementation later (follow DL)

Recommend **for athena** at Sep C&S week

Summary

A lot remains to do for analysis data access APIs

- Already some “legacy” design choices to deal with
- No overarching development thread (that I can see)

Several common themes between Analysis, HLT, Concurrency