Managing Asynchronous Data in ATLAS's Concurrent Framework

John Baines, Tomasz Bold, Paolo Calafiura, Jack Cranshaw, Andrea Dotti, Steven Farrell, Charles Leggett, David Malon, Graeme Stewart, Scott Snyder, Peter Van Gemmeren, Vakhtang Tsulaia, Benjamin Wynne

for the ATLAS Collaboration

ICHEP 2016

► Data that can change during the course of a job, but less frequently than once per Event (beam collision)
  • period for which any piece of data is valid is referred to as an **Interval Of Validity (IOV)**

► Classify into 3 broad types:

  • **Conditions**
    • *eg* high voltages, calibrations, *etc*

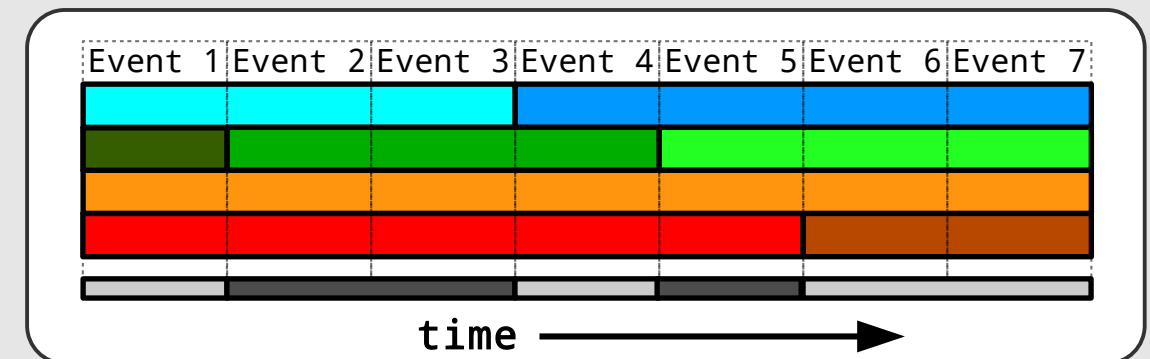 • **Detector Geometry and Alignments**
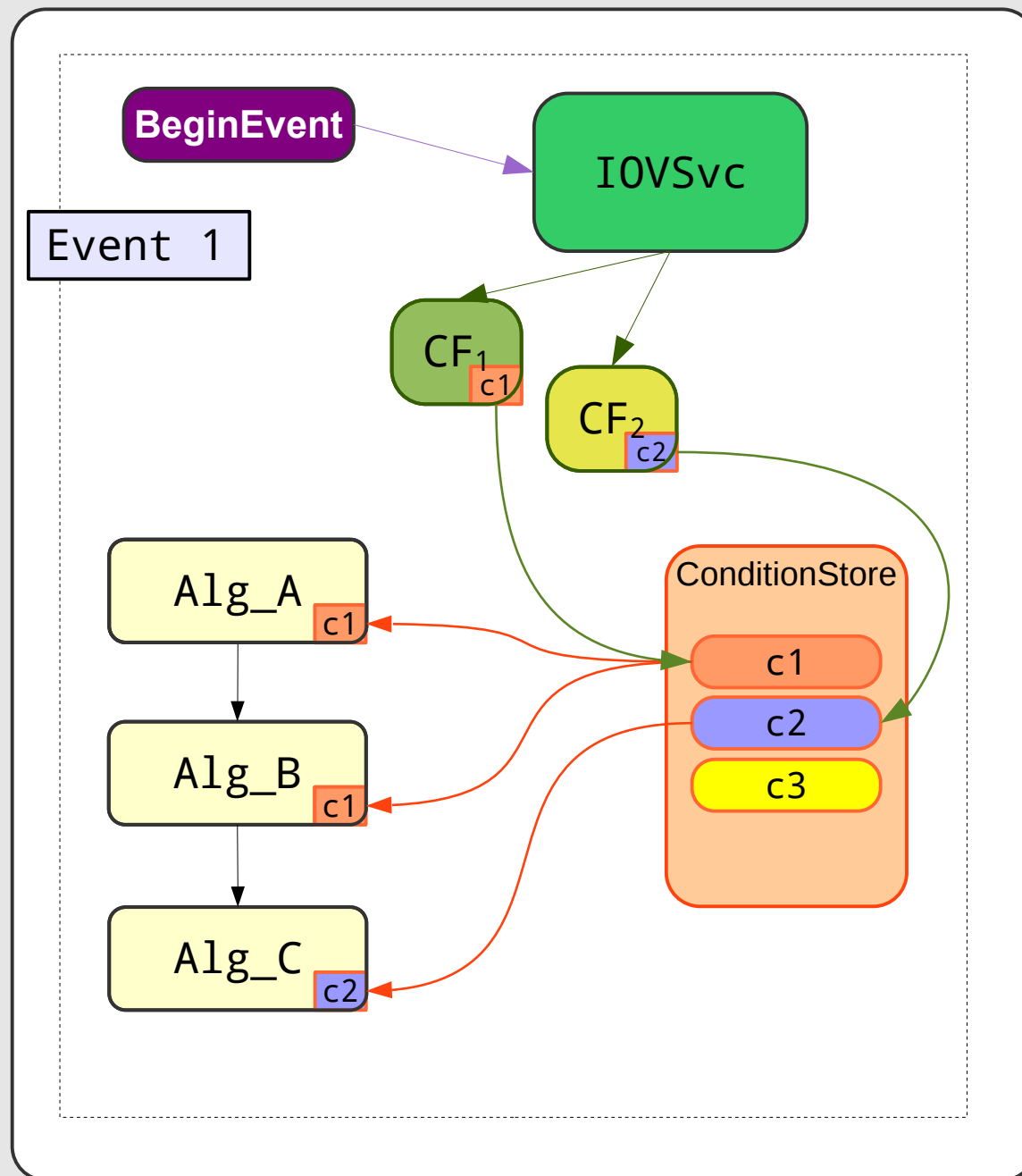   • *eg* position changes

  • **Asynchronous Callbacks (Incidents)**
    • functions that need to be executed at non-predetermined intervals
    • *eg* respond to a file open/close


► These are often inter-related
  • a condition change can trigger an Incident callback

- ▶ All framework elements process data from the same IOV

- ▶ Algorithms are blind to the IOV, retrieve data from **ConditionStore**

- ▶ At the start of every Event, **IOVSvc** checks IOVs, and triggers any necessary updates
  - handled by the Callback Functions
  - Callback Functions are **shared** instances

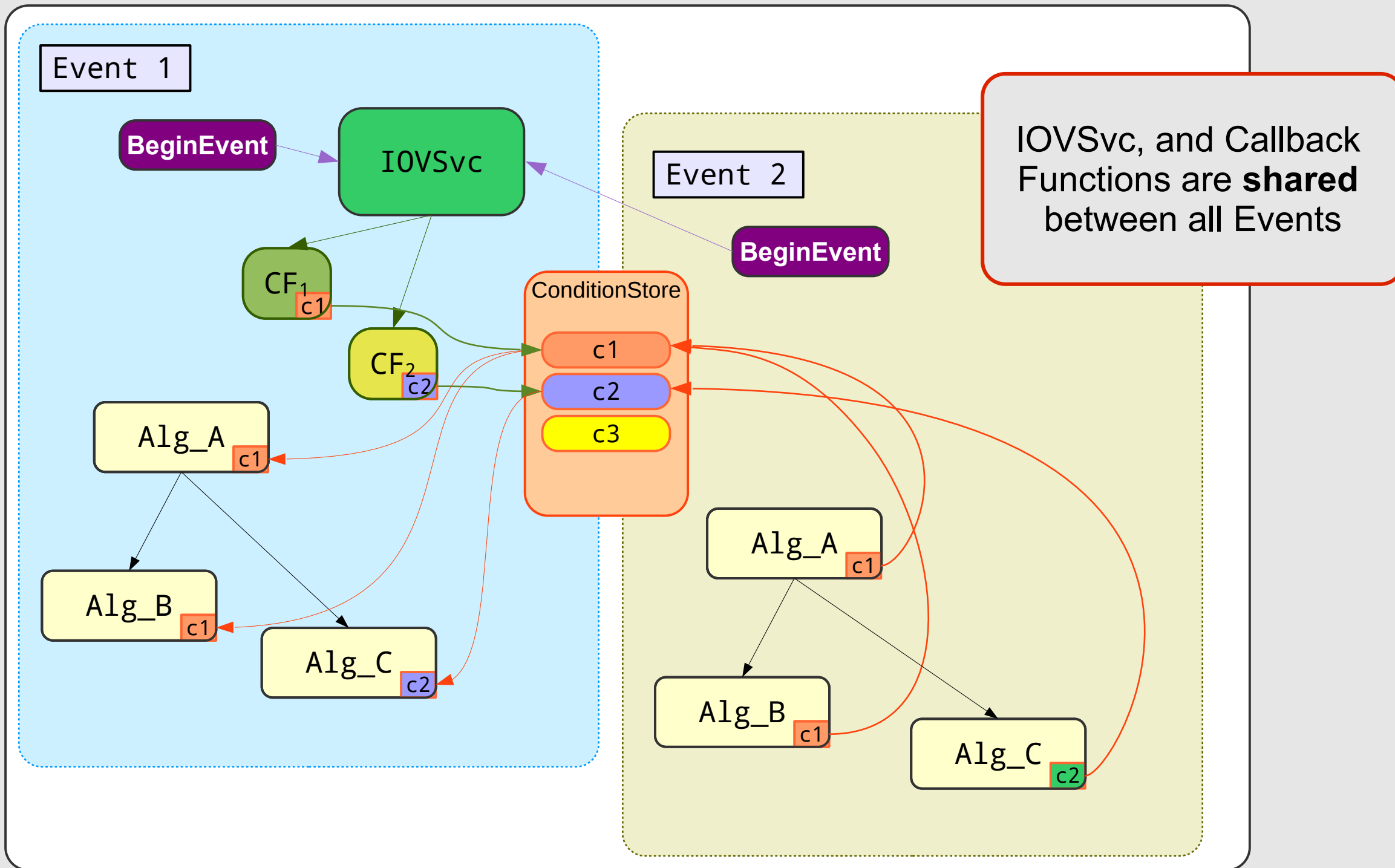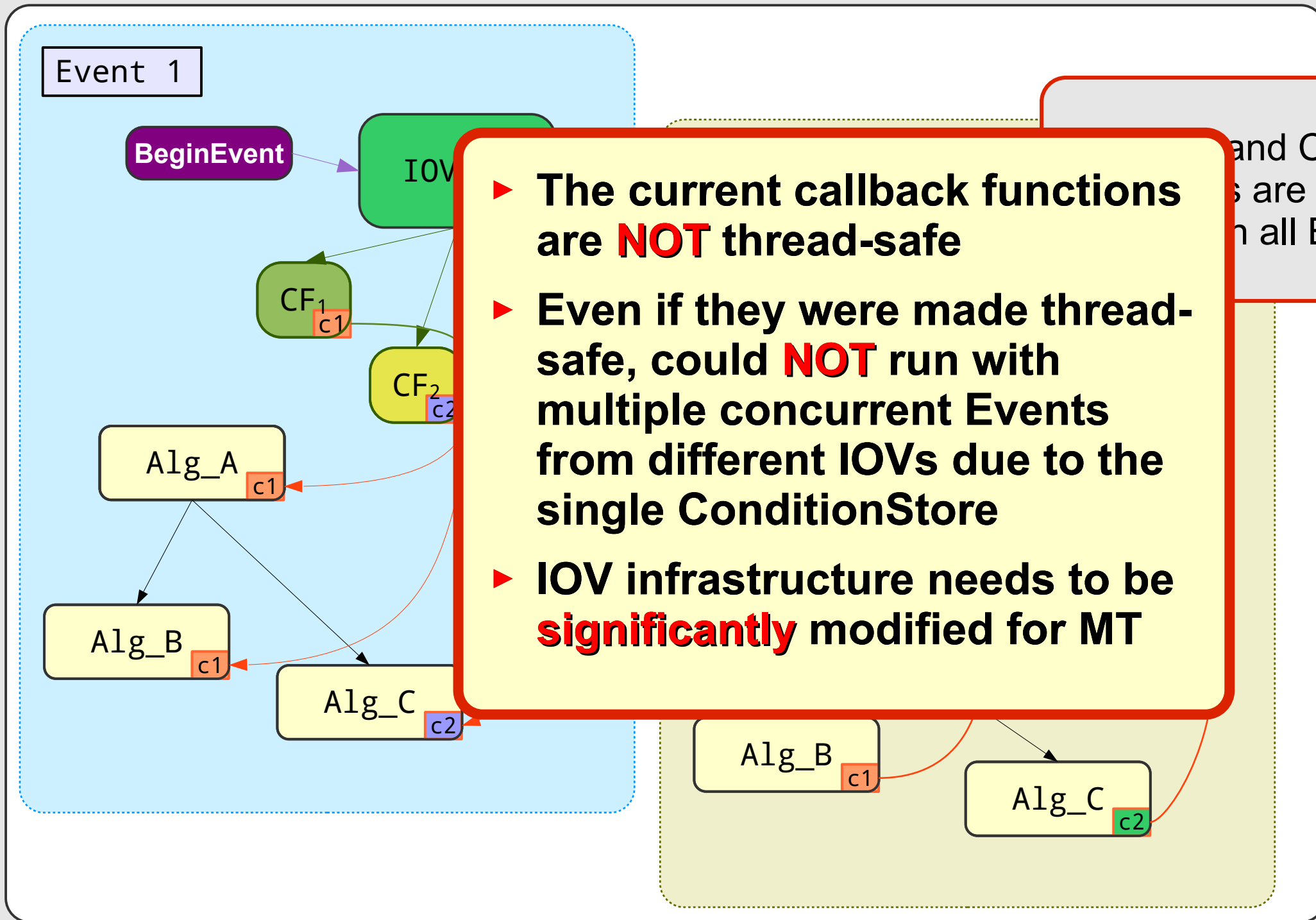- ▶ Only one copy of any Conditions object is maintained in the Store
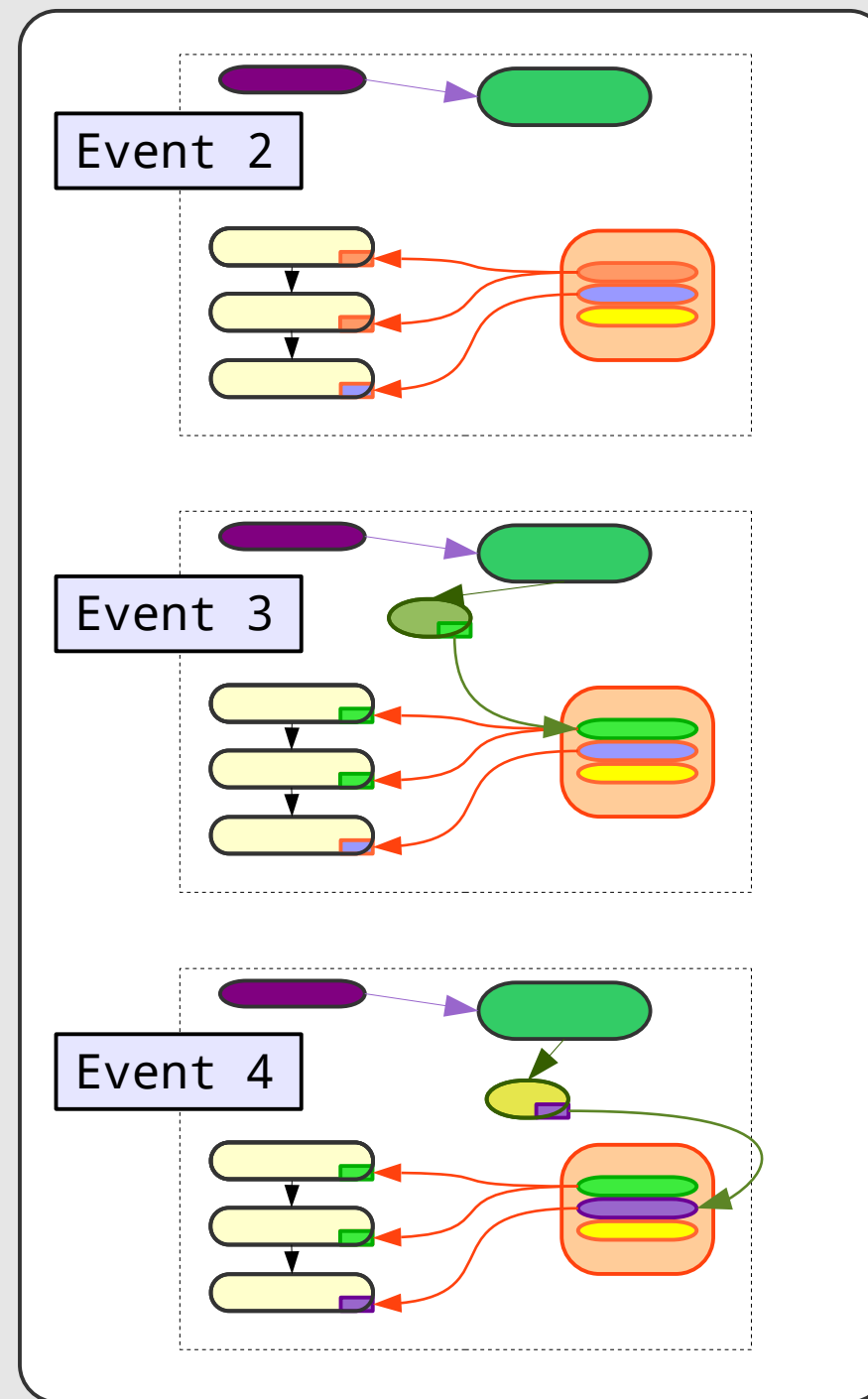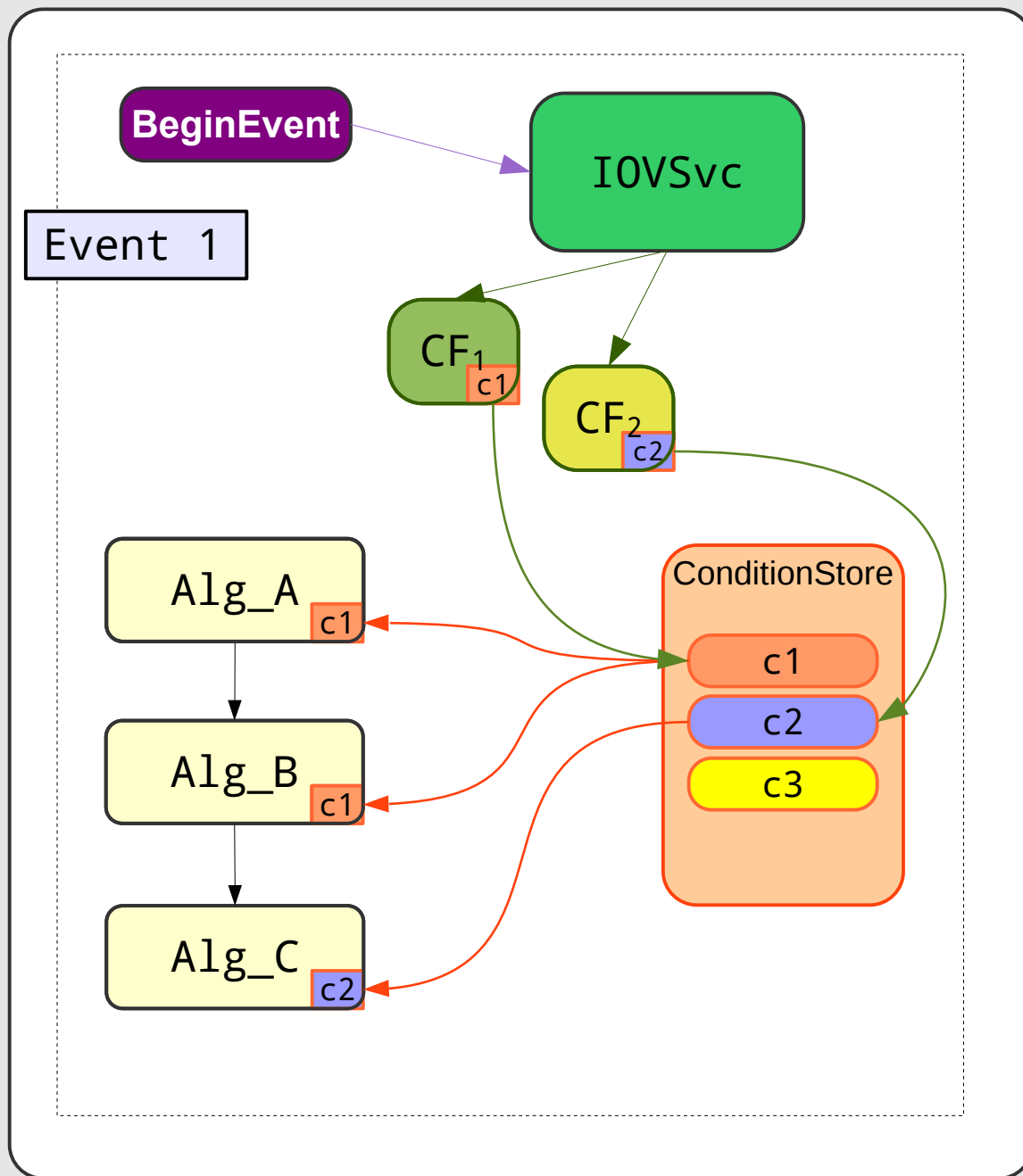
► **AthenaMT**: ATLAS's next generation, multi-threaded reconstruction/simulation framework

- Leverage **multi-threaded** design to minimize memory footprint
  - ATLAS reconstruction is very large
  - ratio of physical memory / CPU is constantly decreasing

- multiple simultaneous Events
- sub-Event concurrency
- each Algorithm processes its Event in its own thread

► **Requirement**: Try to minimize changes to User code

- there's lots and lots of it!
- avoid forcing Users to implement fully thread-safe code by handling most thread-safety issues at the framework / Services level

► **Requirement**: All access to Event data *via* **DataHandles**, which also declare data dependency relationship to the framework

Event 1

BeginEvent

IOV

CF_1 c1

CF_2 c2

Alg_A c1

Alg_B c1

Alg_C c2

...and Callback ...s are **shared** ...n all Events

Alg_B c1

Alg_C c2

> ▸ **The current callback functions are NOT thread-safe**
>
> ▸ **Even if they were made thread-safe, could NOT run with multiple concurrent Events from different IOVs due to the single ConditionStore**
>
> ▸ **IOV infrastructure needs to be significantly modified for MT**

one ConditionStore shared by all Events

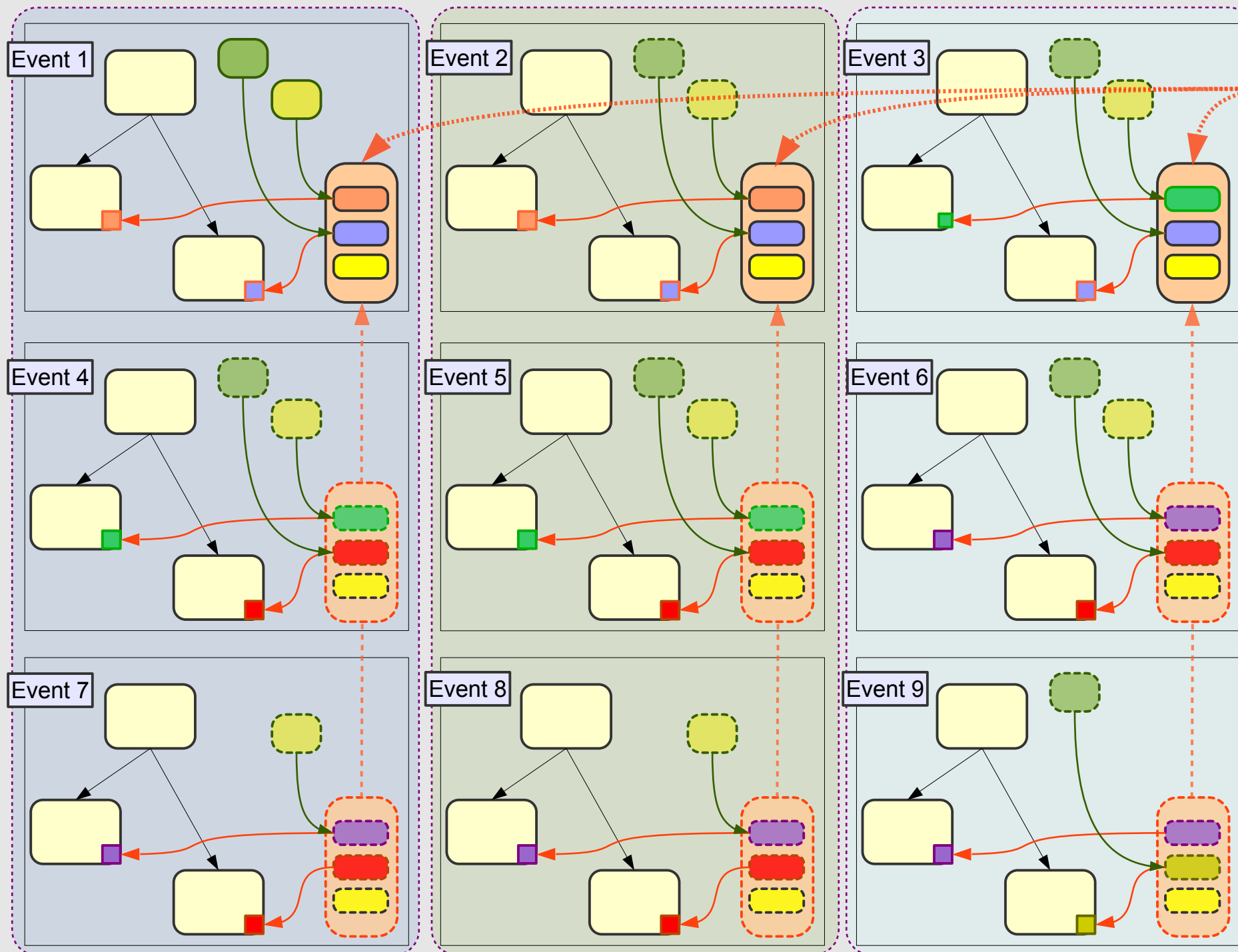Scheduler only **concurrently** process events which have **all** Conditions in the **same** IOV
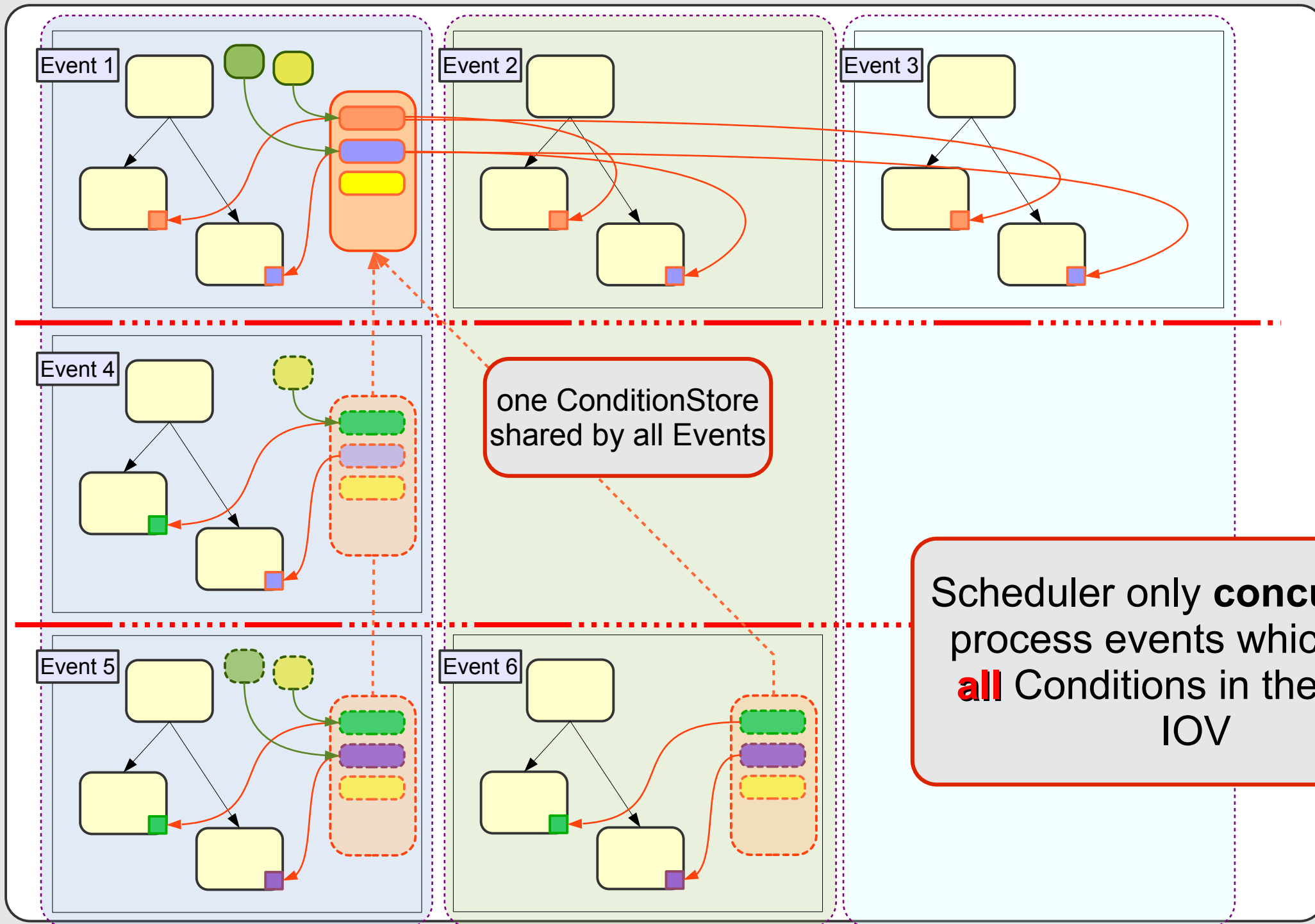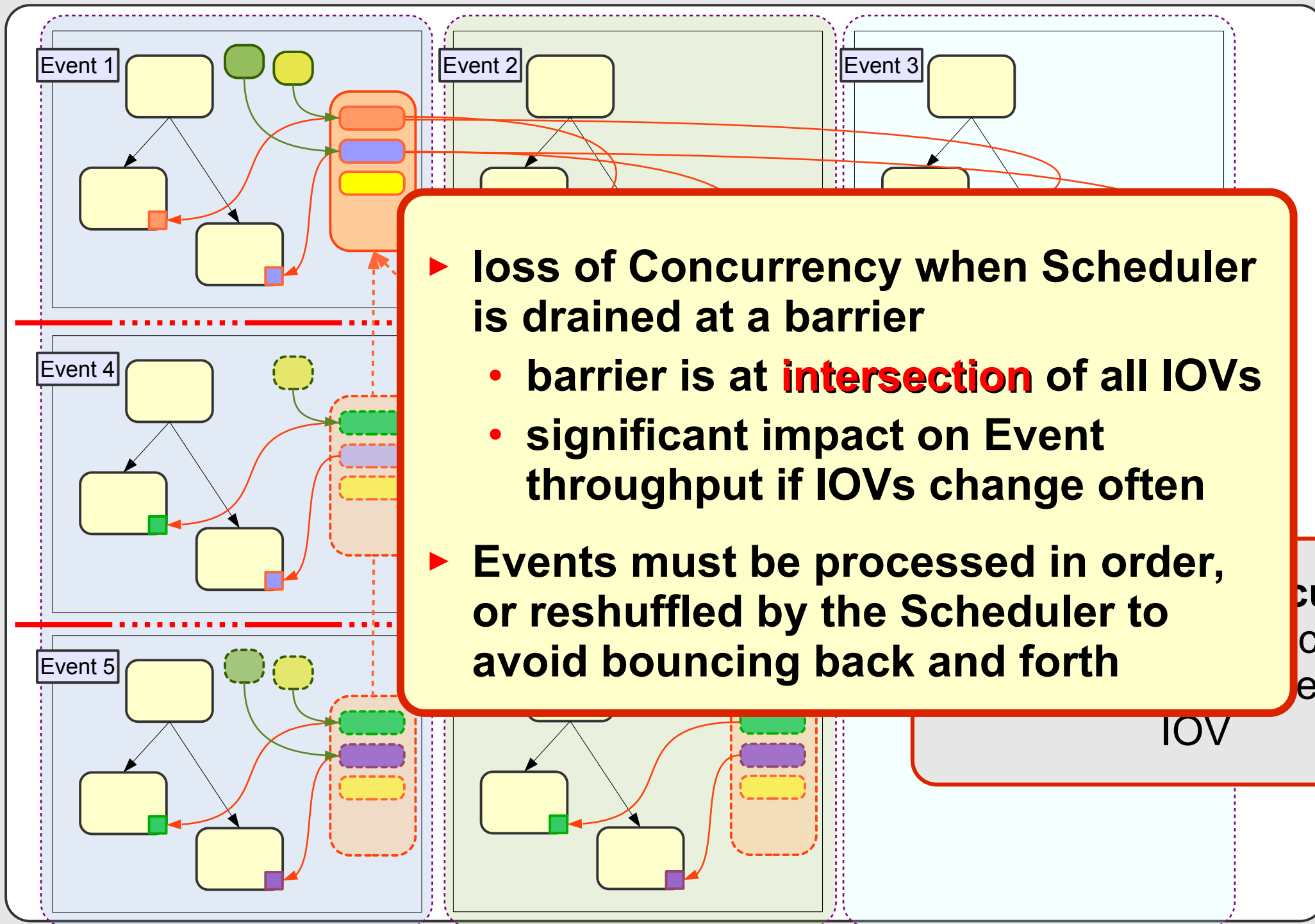
► **loss of Concurrency when Scheduler is drained at a barrier**

- **barrier is at intersection of all IOVs**
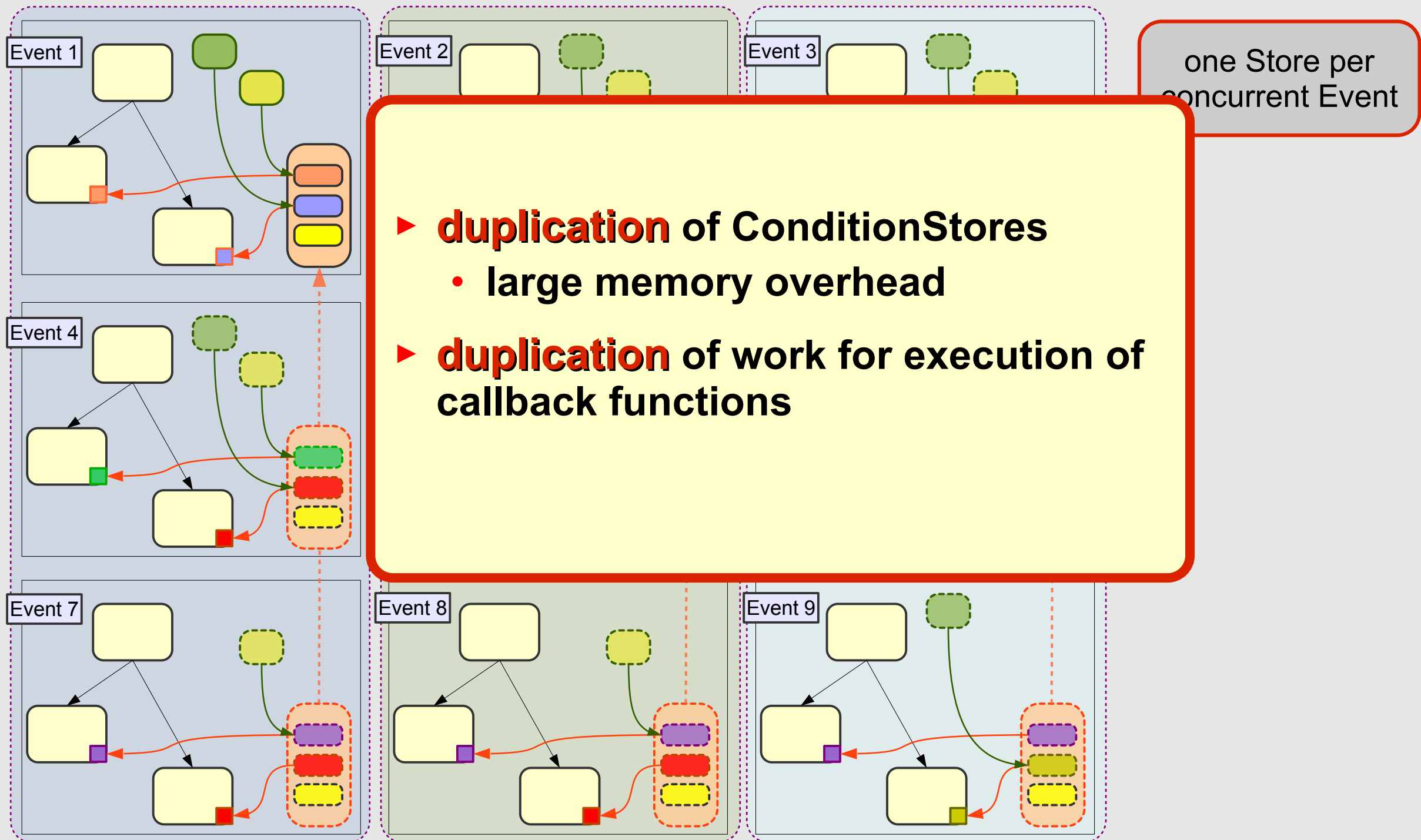- **significant impact on Event throughput if IOVs change often**

► **Events must be processed in order, or reshuffled by the Scheduler to avoid bouncing back and forth**

one Store per concurrent Event

one Store per concurrent Event

▶ **duplication of ConditionStores**
  • **large memory overhead**

▶ **duplication of work for execution of callback functions**

Event 1

Event 2

Event 3

Event 5
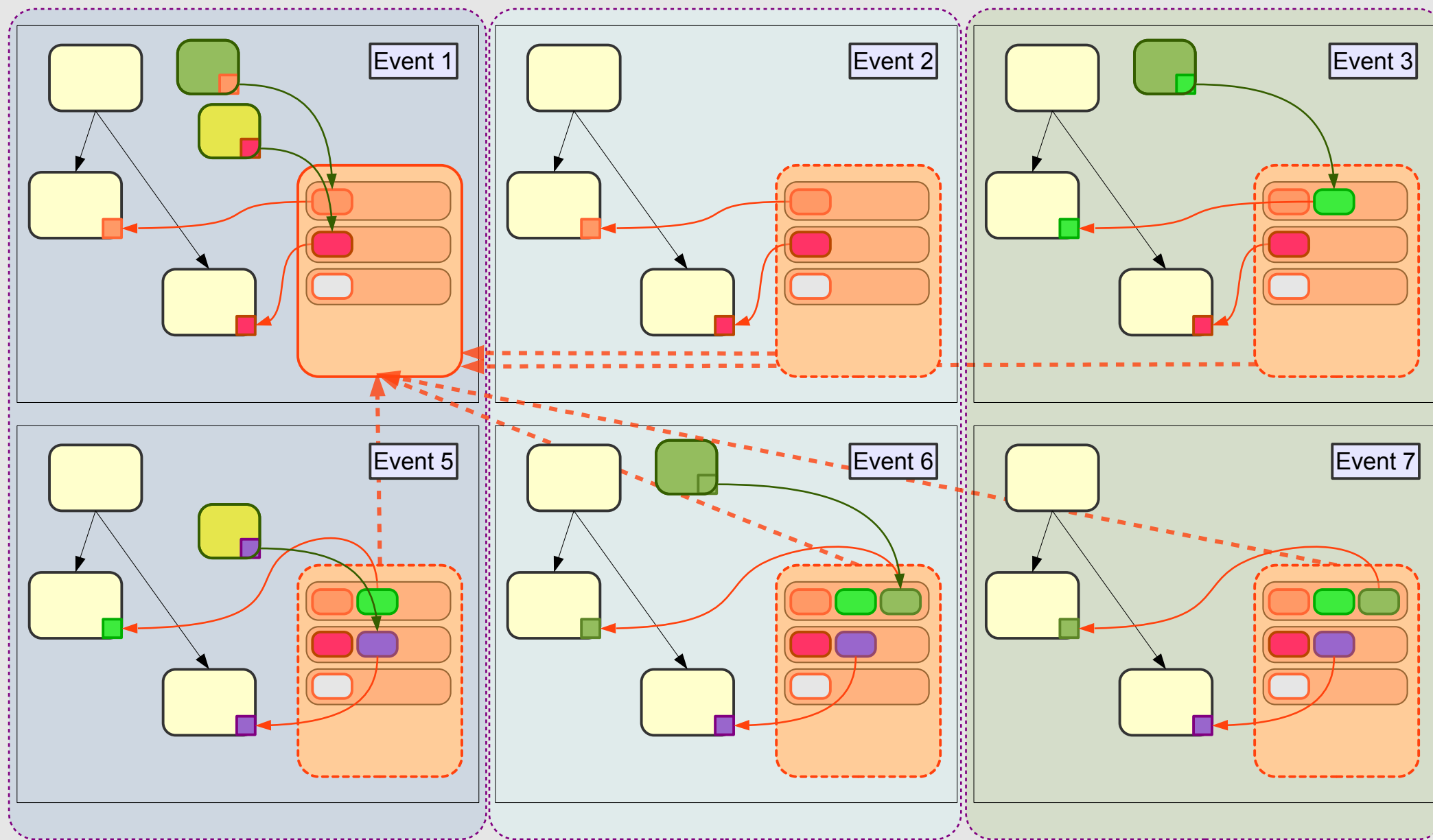
Event 6

Event 7

→ **One** ConditionStore, shared by all Events.

→ Store elements are **ConditionContainers**, with one entry per IOV

→ Data access via **ConditionHandles** that point to appropriate entry

→ Callback Functions become Algorithms, scheduled by framework

Legend:

- **PV** — Physical Volume (basic GeoModel building block)
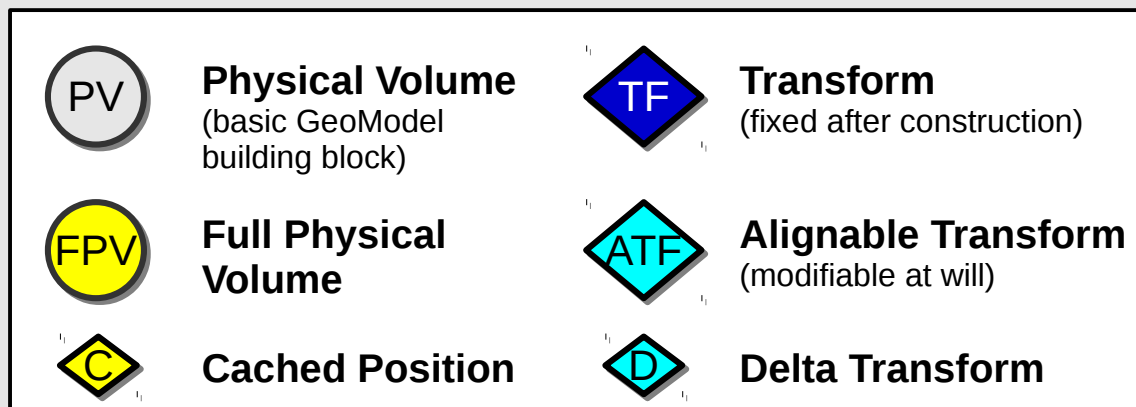- **TF** — Transform (fixed after construction)
- **FPV** — Full Physical Volume
- **ATF** — Alignable Transform (modifiable at will)
- **C** — Cached Position
- **D** — Delta Transform
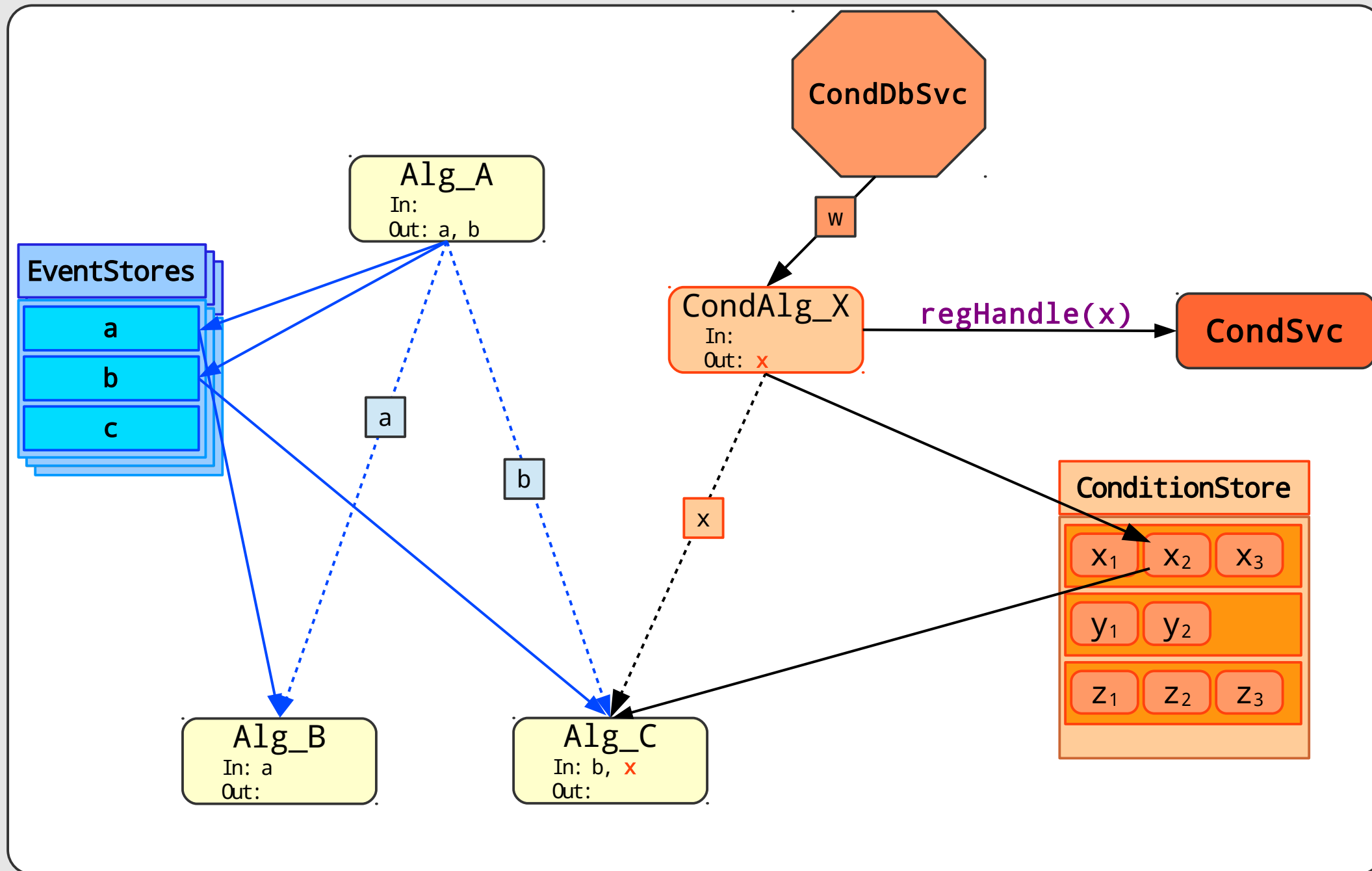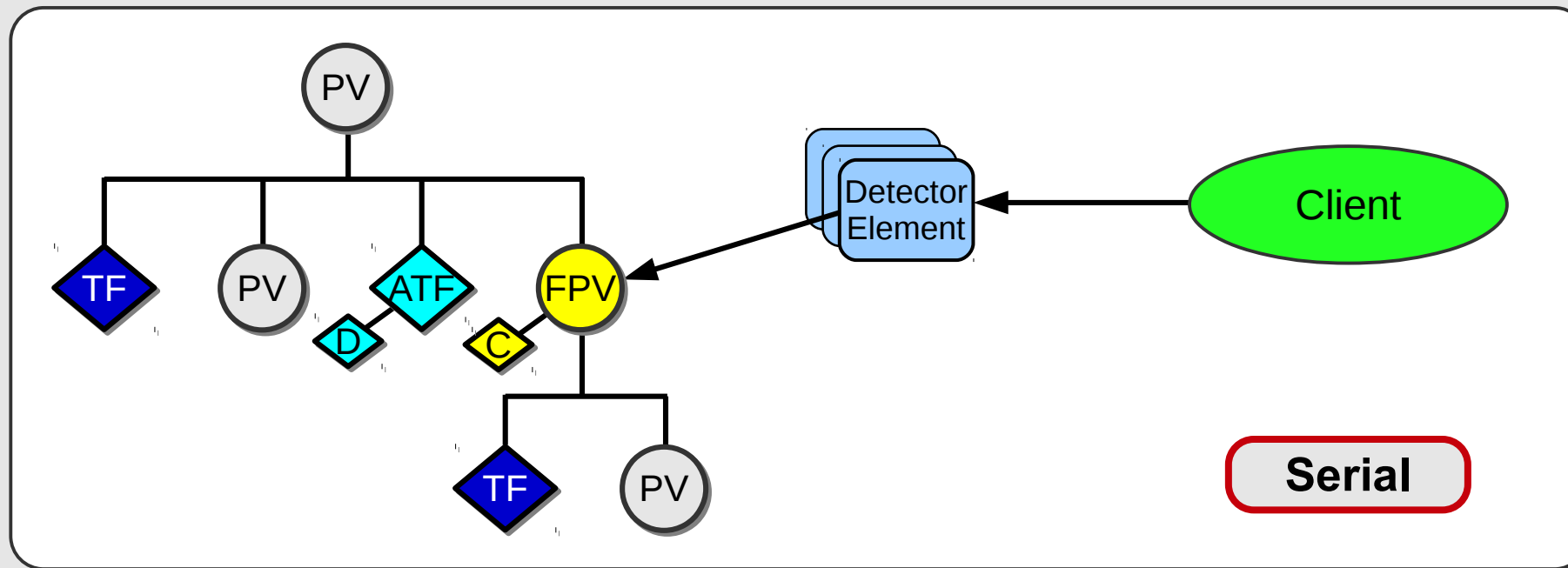
▸ **GeoModel** tree is not exposed to Detector Description clients

▸ Readout geometry layer consists of subsystem specific **Detector Elements**

▸ Each Detector Element has a pointer to **Full Physical Volume**
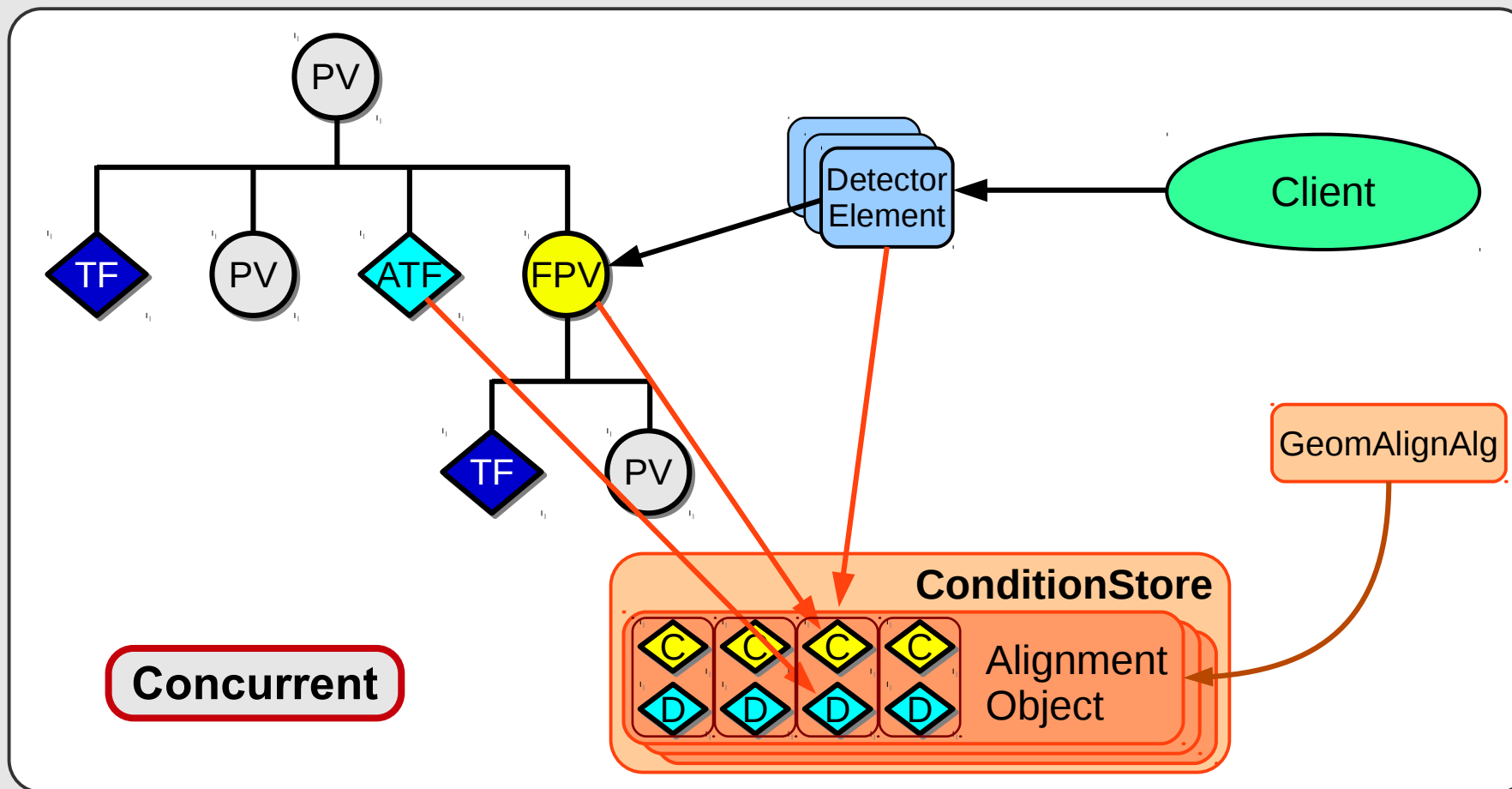
- ▶ The **Alignment Object** is a regular ConditionObject in a ConditionContainer, so it should be handled as any other ConditionObject in AthenaMT
  - Created by a **ConditionAlgorithm** (replacement of current callback function)
  - Accessed from the FPV and ATF via **ConditionHandle**
- ▶ By making Detector Elements aware of the Alignment Objects we can make the transition transparent to Detector Description clients

▶ **IncidentSvc**: manages asynchronous callbacks for clients using an Observer pattern

- *eg*: `BeginEvent`, `EndInputFile`, `MetaDataStop`
- very flexible: callbacks can be triggered at any time
- Clients can be anything: Algorithms, Services, Tools
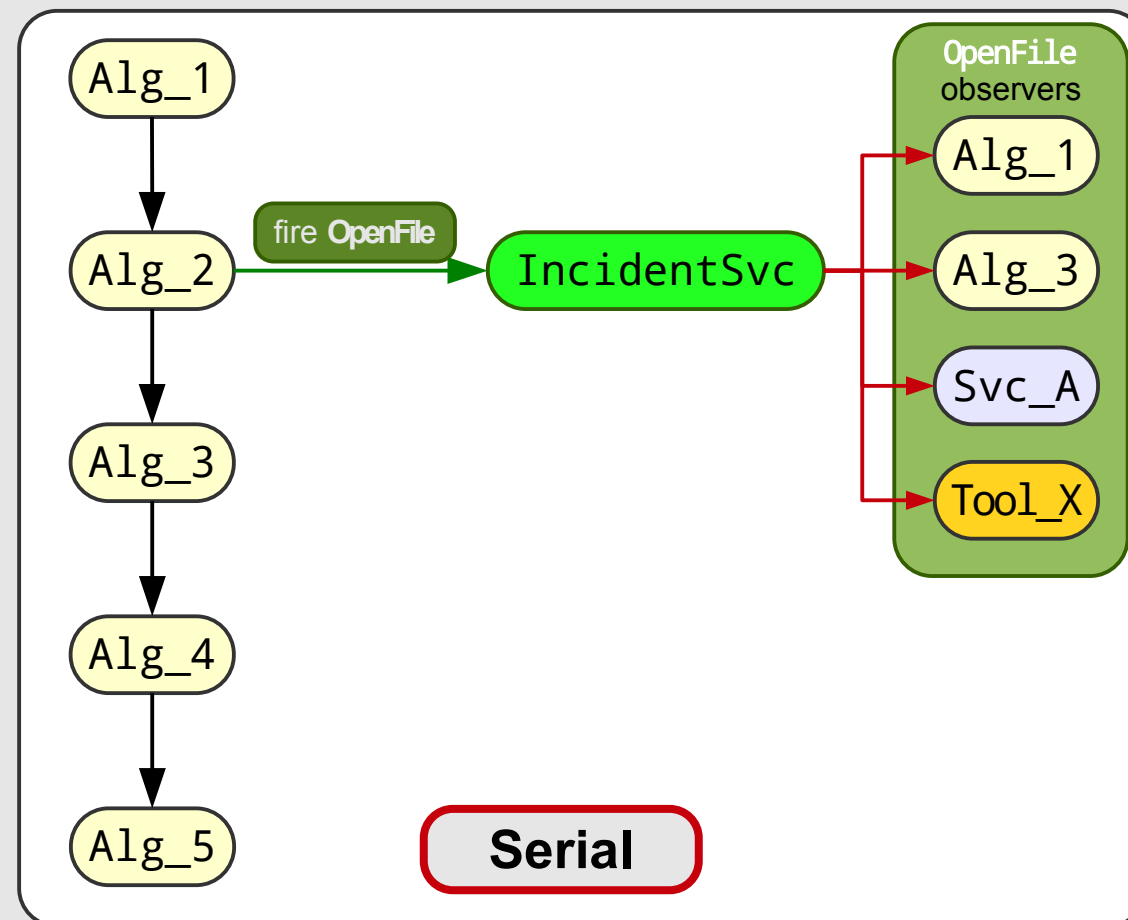
Absolutely disastrous in an environment with multiple concurrent events, and multiple instances of each Algorithm

- ▶ Study: **IncidentSvc** design more flexible than actual usage
  - mostly fired **outside** of the Algorithm processing loop
- ▶ Solution: limit scope of IncidentSvc: Incidents can be re-classified as discrete state changes
  - Incidents become **schedulable**, managed by framework
  - Incident handlers / observers become discrete Algorithms, that interact with Services which are aware of the EventContext
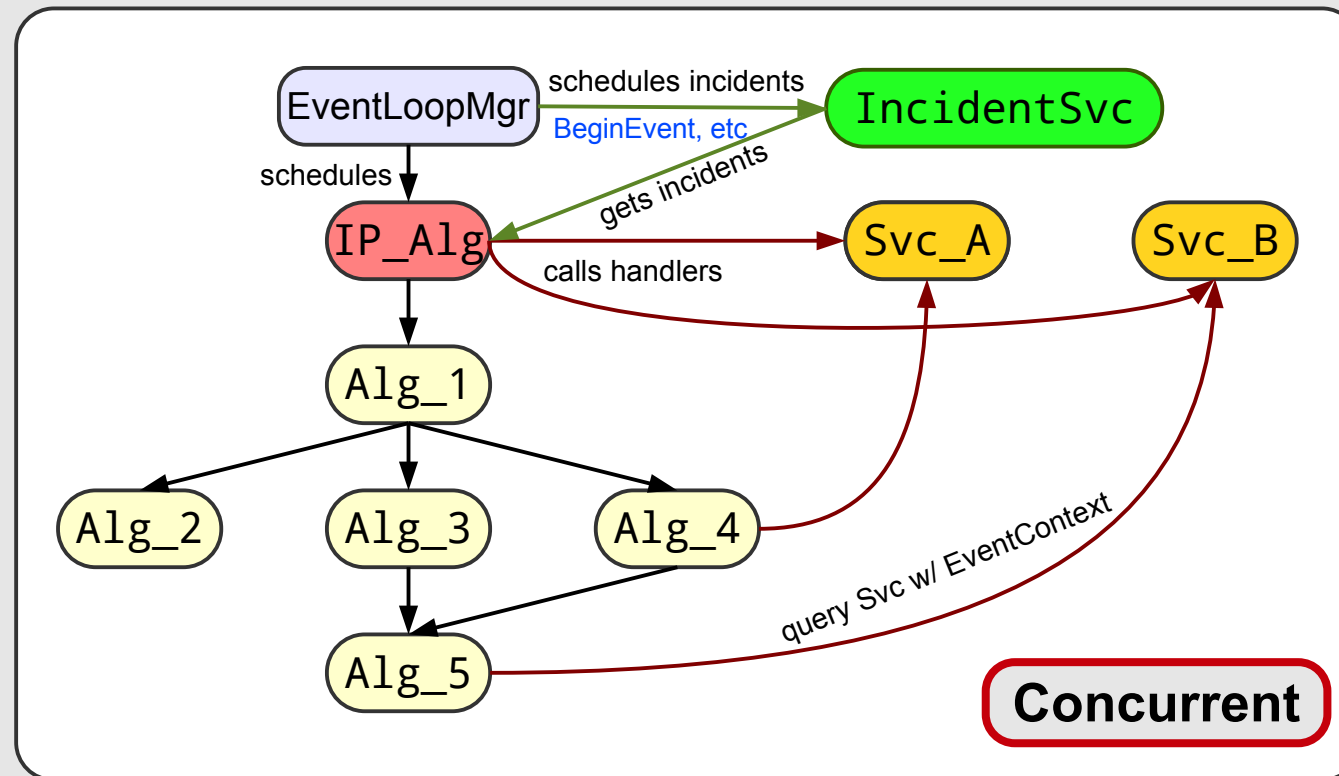
► For ATLAS, managing Asynchronous data in a concurrent environment will require a paradigm shift

- no solution is fully transparent or plug-and-play, unless we choose to sacrifice concurrency and performance
- dealing with multiple threads as well as multiple concurrent events is doubly challenging

► We have been able to minimize impact on User code via strategic modifications at the framework and Service level

► New versions of all three aspects of Asynchronous Data and Event infrastructure have been implemented, and migration of client code is ongoing, in conjunction with the universal migration of Event data access to DataHandles

- so far, migration has been relatively straight-forward, and anticipate finishing by end of 2016

Extras

▶ While a multi-cache store makes optimal use of memory (no duplication of objects), the store will continue to grow with time

▶ Depending on memory constraints, may become necessary to perform garbage collection

- prune ConditionContainers of old, unused entries

▶ Possible pruning techniques:

- only keep N copies
- keep reference count of which entries are in use, purge old entries

- ► **[AbortEvent]**
  - unnamed
- ► **[AfterReseedIncident]**
  - AtRndmGenSvc
- ► **[BeforeFork]**
  - AthenaEventLoopMgr
- ► **[BeginEvent]**
  - AlgContextSvc
  - PerfMonSvc
  - CoreDumpSvc
  - DetectorStore
  - InputMetaDataStore
  - MetaDataStore
  - DataModelCompatSvc
  - TagMetaDataStore
  - StoreGateSvc
  - IOVDbSvc
  - TagInfoMgr
  - IOVSvc.StoreGateSvc
  - AtRndmGenSvc
  - ToolSvc.CaloLumiBCIDToolDefault
  - ToolSvc.InDetMergedPixelsTool
  - InDetSCT_FlaggedConditionSvc
  - InDetSCT_ByteStreamErrorsSvc
  - ToolSvc.SolenoidalIntersector
  - ToolSvc.egammatracktovertex
  - ToolSvc.Reco::TrackToVertex
  - ToolSvc.CaloAffectedTool
  - ToolSvc.softetracktovertex
  - ToolSvc.DefaultJetVtxTrackHelper
  - ToolSvc.MissingCellListTool
  - ToolSvc.ClusterRhoKt4EM
  - ToolSvc.ClusterRhoKt4LC
  - ToolSvc.ClusterRhoKt6EM
  - ToolSvc.ClusterRhoKt6LC
  - UserDataSvc
  - ToolSvc.tauRec_JetVtxTrkHelper
  - ThinningSvc
  - ToolSvc.CaloAffectedToolDefault
  - AlgContextSvc

- ► **[LastInputFile]**
  - ToolSvc.ByteStreamMetadataTool
  - MetaDataSvc
  - CreateLumiBlockCollectionFromFile
- ► **[ReseedIncident]**
  - AtRndmGenSvc
- ► **[BeginInputFile]**
  - IoComponentMgr
  - ToolSvc.IOVDbMetaDataTool
  - ToolSvc.ByteStreamMetadataTool
  - MetaDataSvc
  - TagInfoMgr
  - CreateLumiBlockCollectionFromFile
- ► **[BeginOutputFile]**
  - IoComponentMgr
- ► **[BeginRun]**
  - CoreDumpSvc
  - IOVDbSvc
  - TagInfoMgr
  - IOVSvc.StoreGateSvc
  - TileInfoLoader
  - ToolSvc.LArADC2MeVToolDefault
  - EmTowerBldr.LArEmTwrBldr
  - LArAffectedRegionAlg
  - SCT_CablingSvc
  - InDetSCT_ByteStreamErrorsSvc
  - ToolSvc.InDetTRT_DriftFunctionTool
  - InDetTRTActiveFractionSvc
  - RegSelSvc
  - CmbTowerBldr.TileCmbTwrBldr
  - CmbTowerBldr.LArCmbTwrBldr
  - CmbTowerBldr.LArFCalCmbTwrBldr
- ► **[CheckIOV]**
  - IOVSvc.StoreGateSvc
- ► **[BeginTagFile]**
  - MetaDataSvc

- ► **[EndInputFile]**
  - ToolSvc.ByteStreamMetadataTool
  - MetaDataSvc
  - CreateLumiBlockCollectionFromFile
- ► **[EndOfBeginRun]**
  - IOVDbSvc
- ► **[EndTagFile]**
  - MetaDataSvc
- ► **[EndEvent]**
  - AlgContextSvc
  - PerfMonSvc
  - CoreDumpSvc
  - DetectorStore
  - InputMetaDataStore
  - MetaDataStore
  - TagMetaDataStore
  - StoreGateSvc
  - xAODMaker::EventFormatSvc
  - AtRndmGenSvc
  - CoolHistSvc
  - ToolSvc.InDetMergedPixelsTool
  - ToolSvc.InDetPixelClusterOnTrackTool
  - ToolSvc.InDetBroadPixelClusterOnTrackTool
  - ToolSvc.MuonTruthSummaryTool
  - ToolSvc.MuonLayerHoughTool
  - MuonStationIntersectSvc
  - ToolSvc.MuPatHitTool
  - ToolSvc.MuPatCandidateTool
  - ToolSvc.egPixelClusterOnTrackTool
  - THistSvc
  - ToolSvc.JetTrackToVertexAssociator
  - ToolSvc.tauRec_JetTrackToVertexAssociator
  - DecisionSvc

- ► **[FirstInputFile]**
  - ToolSvc.IOVDbMetaDataTool
  - ToolSvc.ByteStreamMetadataTool
  - MetaDataSvc
- ► **[MetaDataStop]**
  - StreamESD
  - StreamAOD
- ► **[StoreCleared]**
  - CoreDumpSvc
  - IOVDbSvc
  - GeoModelSvc.MuonDetectorTool
  - InDetPixelClusterization
  - InDetSCT_Clusterization
  - ToolSvc.TrackExtrapolatorToCalo
  - ToolSvc.extrapolMuonInCaloTool
  - JetGlobalEventSetup
- ► **[ModuleLoaded]**
  - ClassIDSvc
- ► **[PostFinalize]**
  - PerfMonSvc