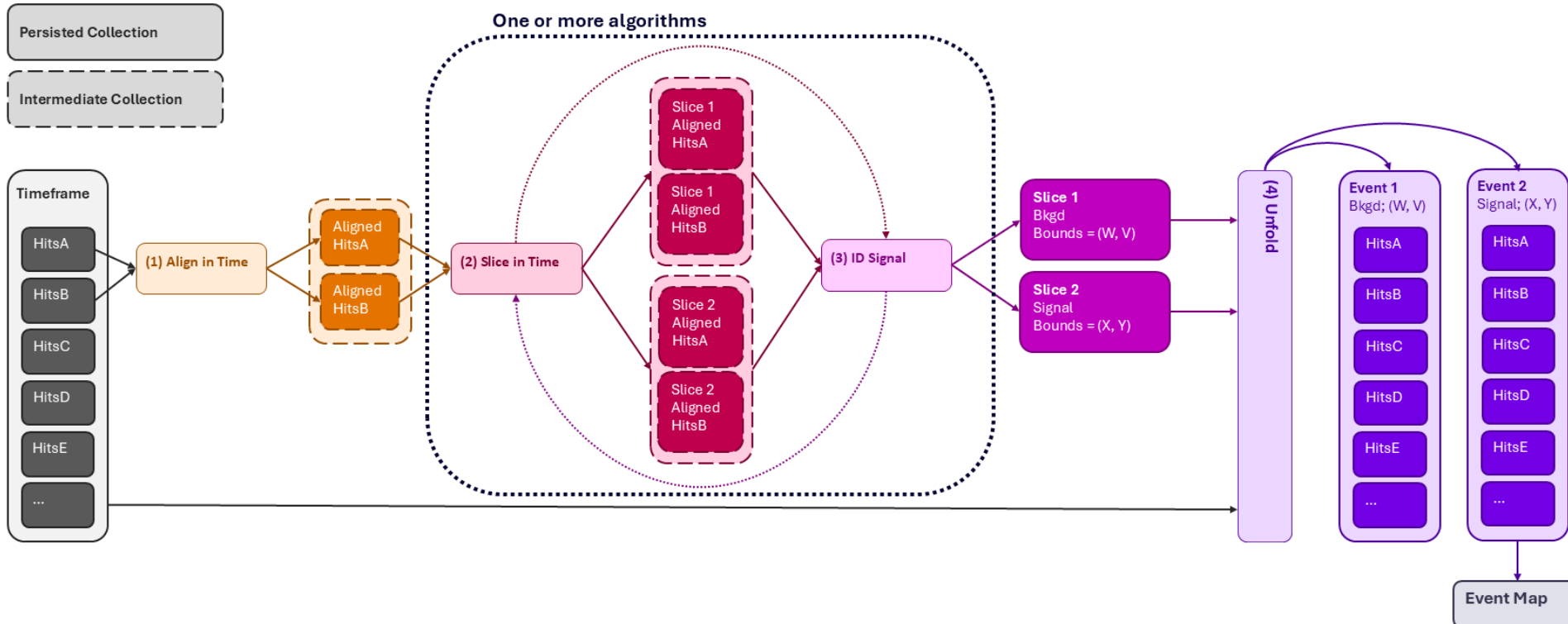
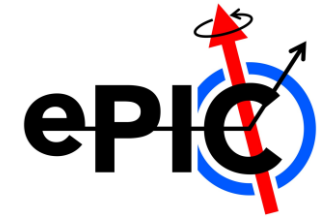
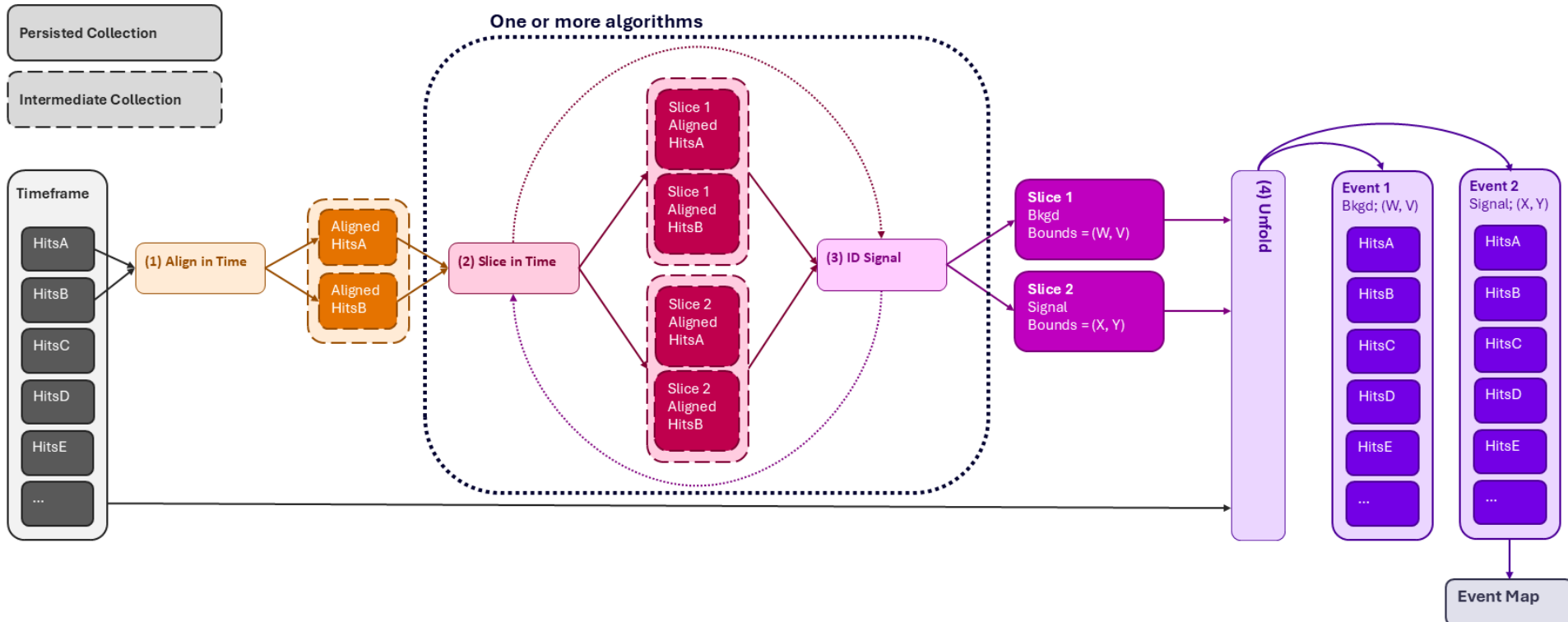


Timeslice Strawman V2 | Minimum-Viable Product



- **Above:** updated timeslice strawman after feedback
 - Dotted lines indicate areas to evolve after MVP (e.g. iterative slicing, separating pieces of splitting algo)
 - Dashed lines indicate intermediate collections
 - Slice signal/bkgd ID + bounds container still TBD
- **Minimum Viable Product (MVP):** aligning & splitting are algorithm(-like)
 - Alignment happens only on collections needed for splitting
 - Splitting (slicing + signal/bkgd ID) happens in one algo, ingesting only needed collections & emitting slice bounds/IDs
 - Unfolding happens outside splitting algo

Timeslice Strawman V2 | Target Interfaces



(1) Align in Time

```
app→Add(JOmnifactoryGeneratorT<HitTimeAlignment>(  
    "AlignedHitsA",  
    {"HitsA", "HitsB"},  
    {"AlignedHitsA", "AlignedHitsB"},  
    {}, JEventLevel::Timeslice  
));
```

(2/3) Slice in Time + ID Signal

```
app→Add(JOmnifactory<MultiplicitySlicer>(  
    "SliceBounds",  
    {"AlignedHitsA", "AlignedHitsB"},  
    {"SliceBounds"},  
    {}, JEventLevel::Timeslice  
));
```

(4) Unfold

```
app→Add(JSplitter(  
    {"SliceBounds"}, // how to slice  
    {"HitsA", "HitsB", "HitsC", ...}, // what to slice  
));
```