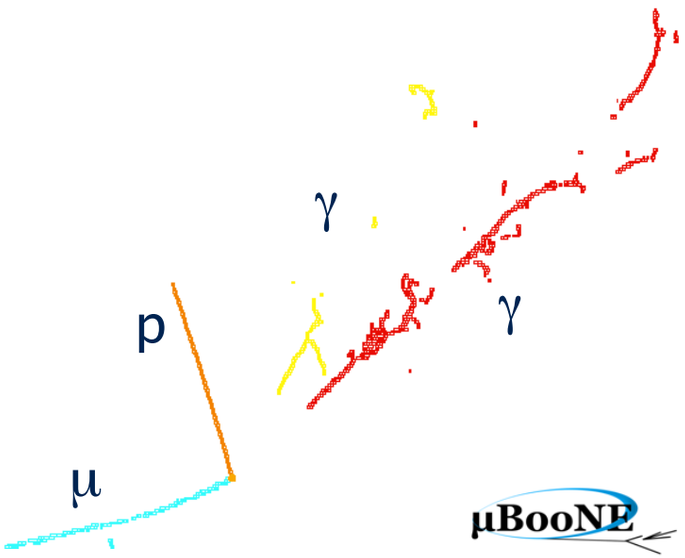


Pandora Talk 1: Overview

J. S. Marshall for the Pandora Team

MicroBooNE Pandora Workshop

July 11-14th 2016, Cambridge

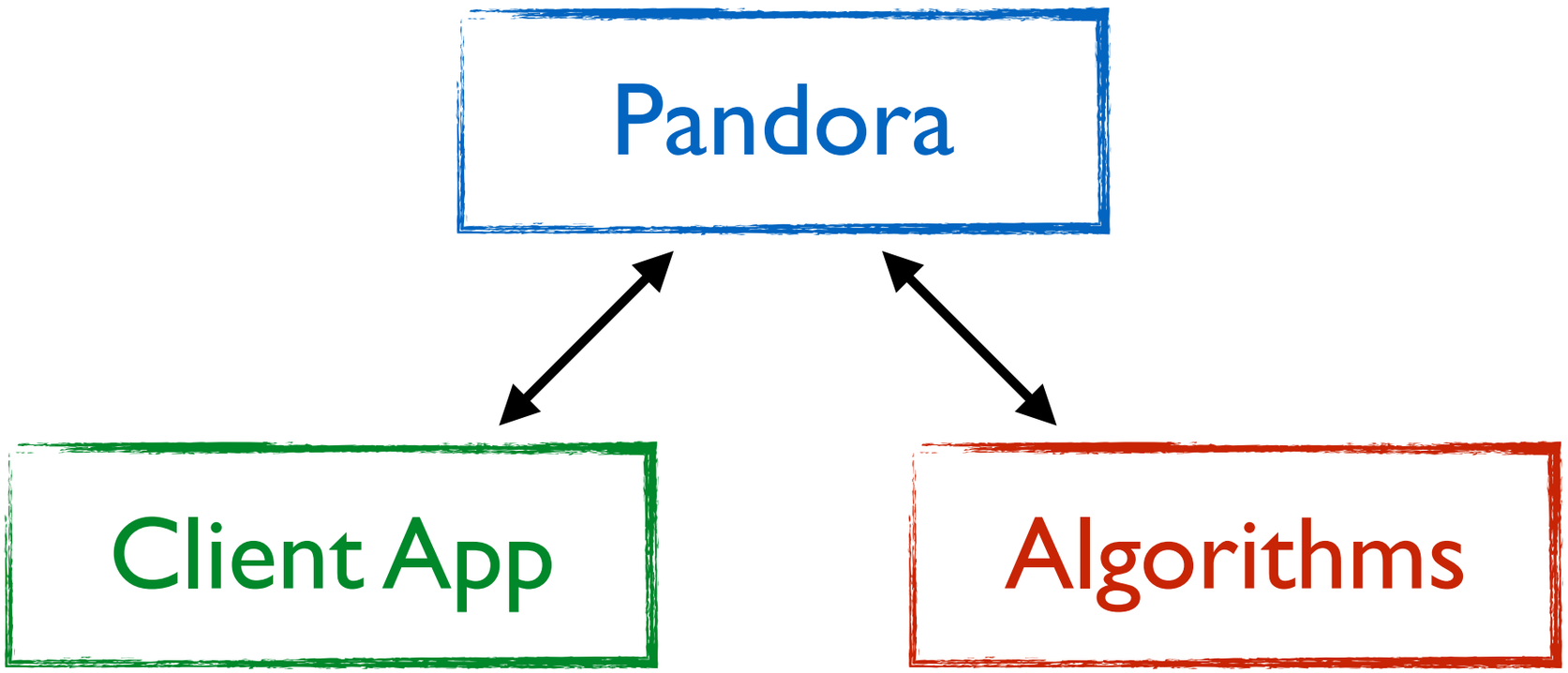
Pandora Overview

- Have discussed the strengths of the multi-algorithm approach to pattern recognition:
  - Focus now on how the Pandora project has tried to put it into practice
- May have heard some of this before, but usually at rather a high, abstract level:
  - Over the next few days discuss every aspect in detail and try it all out yourself!

In this short talk, will try to provide (another high-level!) overview of development with Pandora and place all the upcoming talks and exercises in context.

Pandora Design

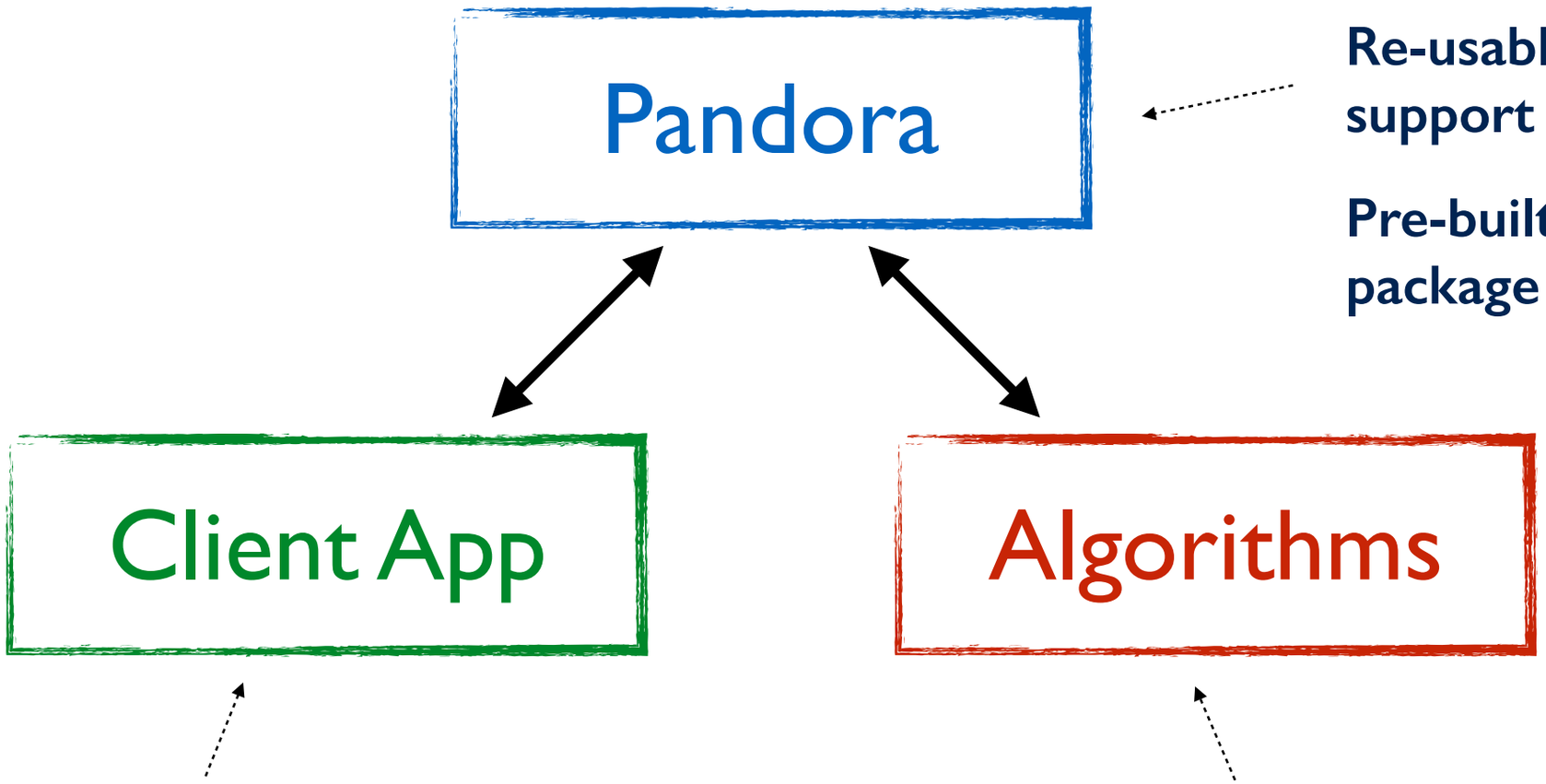
The Pandora multi-algorithm approach:



Pandora design requirements:

1. **Easy to provide the building-blocks that define a pattern recognition problem.**
2. **Logic required to solve pattern recognition problems cleanly implemented in algorithms.**
3. **Operations to access or modify building-blocks requested by algs, performed by Pandora.**

Pandora in LArSoft



pandora

Re-usable libraries with APIs, support multi-alg approach

Pre-built as an external package by LArSoft managers

larpandora

Producer module, provides translation LArSoft↔Pandora

Hosted on Redmine, build with cetbuildtools

larpandoracontent

80+ algorithms and tools, specifically for LAr TPC usage

Newly hosted on Redmine, can now build with cetbuildtools

Client App

- **A client application is responsible for controlling Pandora pattern recognition: it creates the Pandora instance(s) and uses the Pandora APIs to request services.**
- It registers algorithm factories, giving Pandora instances the ability to instantiate specific algorithms, and it provides the algorithm configuration via an XML file.
- Each event, it asks Pandora to create the input 'building blocks' (e.g. Hits and, optionally, MCParticles) to describe an event and it receives the final output Particles.

- To create an input building blocks must provide a list of simple parameters: energy, position, etc.
- Algorithms access information stored in building blocks but do not need to know how information was obtained.
- **Client application isolates Pandora algorithms from host framework.**

Algorithm Pseudocode description of a client application for LAr TPC event reconstruction in a single drift volume

```
1:  procedure MAIN
2:      Create a Pandora instance
3:      Register Algorithms and Plugins
4:      Ask Pandora to parse XML settings file
5:      for all Events do
6:          Create CaloHit instances
7:          Create MCParticle instances
8:          Specify MCParticle-CaloHit relationships
9:          Ask Pandora to process the event
10:         Get output PFOs and write to file
11:         Reset Pandora before next event
```

Client App

Talks:

2. The Pandora client application: larpandora (very next talk).

8. The Pandora output to LArSoft

Exercises:

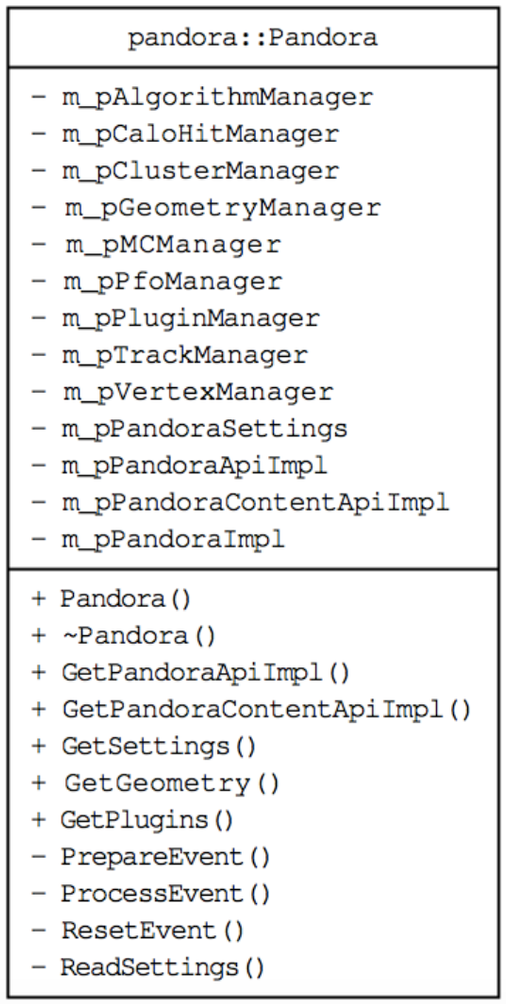
1. Run larpandora, turn on Pandora visualisation and write input events to Pandora formats.

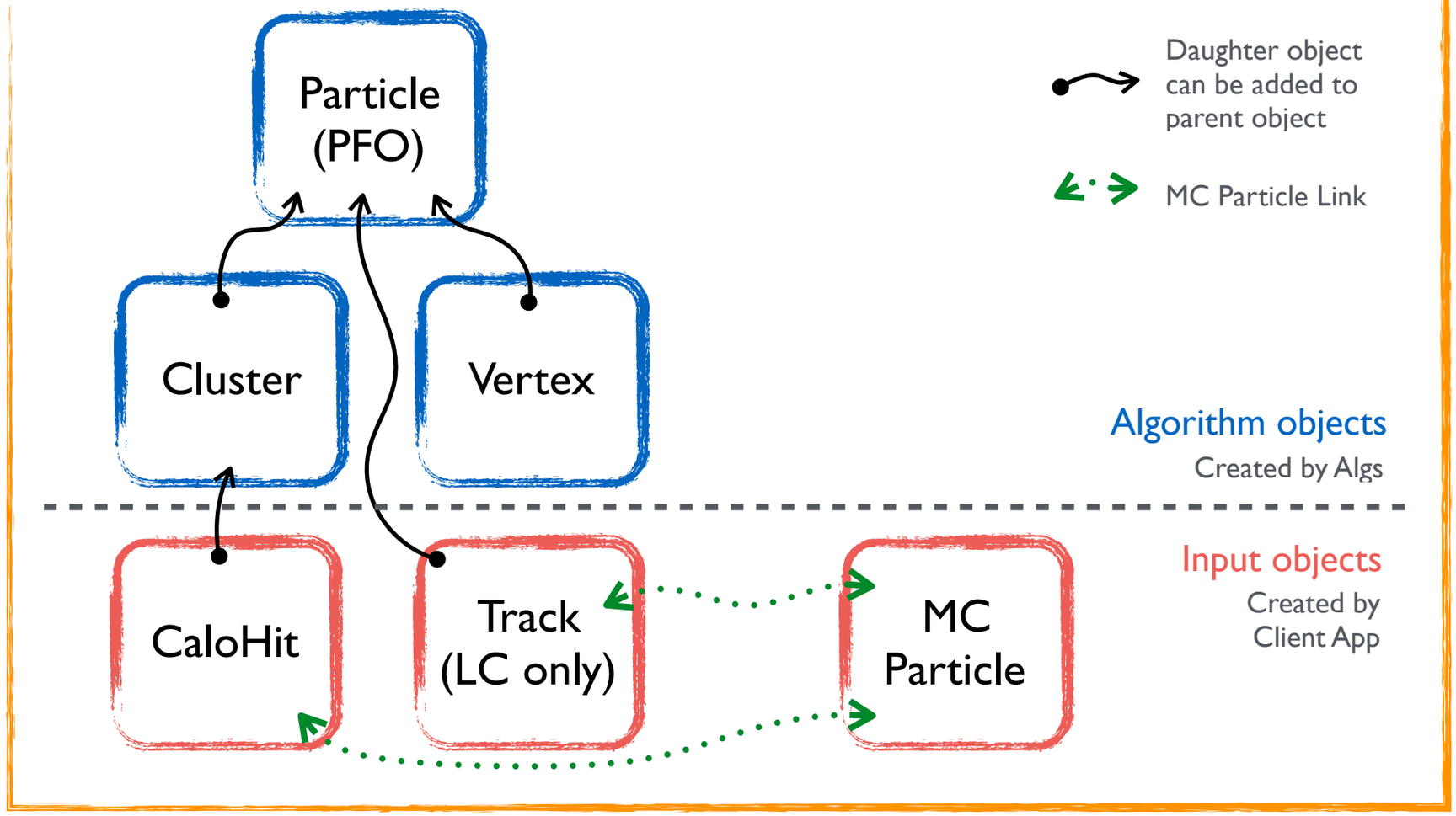
2. Build and develop a very simple client app in the Pandora standalone development environment.

6. Read Pandora outputs, persist in LArSoft EDM and use/process in an analyzer and/or filter.

Pandora

- The Pandora SDK provides a comprehensive **Event Data Model** (EDM) for managing pattern recognition problems. Instances of objects in the EDM are owned by **Pandora Managers**.
- The object instances are stored in named lists and the Managers are able to create new objects, delete objects, create and save new lists and move objects between lists.
- The Managers provide a complete set of low-level operations that allow the high-level operations requested by pattern recognition algorithms to be satisfied.





Pandora

Talks:

3. The Pandora SDK: Algorithms, Tools, Event Data Model and XML configuration

Exercises:

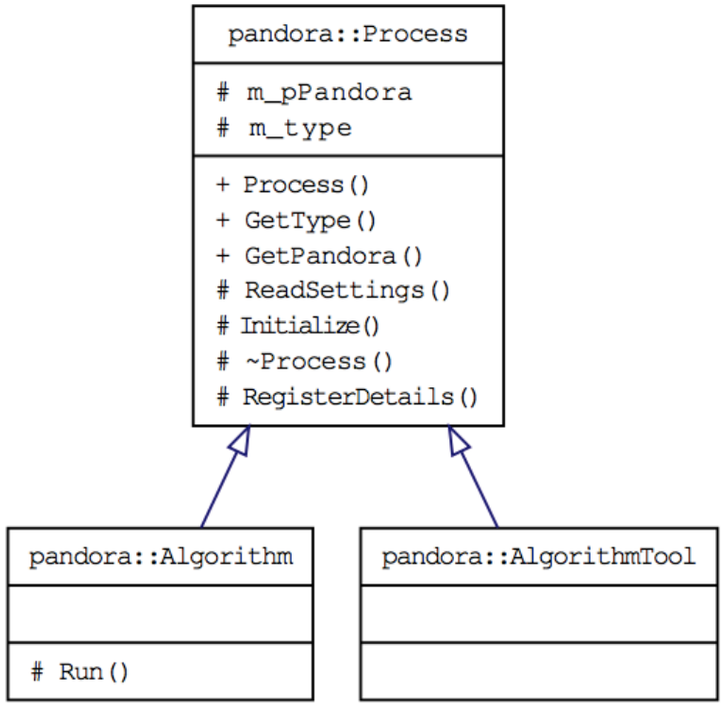
All exercises, 1-9, involve requesting services from Pandora

Publication:

EPJC75:439 or arXiv:1506.05348

Algorithms

- Pandora algorithms contain the step-by-step instructions for finding patterns in the provided data: provide the "brain power".
- Algorithms can use APIs to access Pandora "content" and to ask Pandora to make new objects or modify existing objects.
- **Each API implementation fully tested** so that it can be used with total confidence. Exception handling ensures robustness.



Algorithm 1 Cluster creation pseudocode. The logic determining when to create new Clusters and when to extend existing Clusters will vary between algorithms.

```
 1: procedure CLUSTER CREATION
 2:     Create temporary Cluster list
 3:     Get current CaloHit list
 4:     for all CaloHits do
 5:         if CaloHit available then
 6:             for all newly-created Clusters do
 7:                 Find best host Cluster
 8:             if Suitable host Cluster found then
 9:                 Add CaloHit to host Cluster
10:             else
11:                 Add CaloHit to a new Cluster
12:     Save new Clusters in a named list
```

Algorithm 2 Cluster merging pseudocode. The logic governing the identification of suitable parent Clusters and daughter Clusters will vary between algorithms.

```
1: procedure CLUSTER MERGING
2:     Get current Cluster list
3:     for all Clusters do
4:         if Cluster is suitable parent then
5:             for all Clusters do
6:                 Find best daughter Cluster
7:             if Suitable daughter Cluster found then
8:                 Merge daughter Cluster into Parent
```

Algorithms

- Algorithms are structured around key operations and can be written in simple pseudo-code form. Common algorithms, with associated design patterns, include:
  - `Creating Clusters (containers of Hits)`
  - `Refining Clusters (Cluster merging or Cluster splitting)`
  - `Creating Particles (containers of Clusters, Tracks and Vertices)`
  - `Refining Particles (Particle merging or Particle splitting)`

- Developers can focus on providing Boolean logic to drive operations; typically determined by investigating event topology. Helps to concentrate thoughts/ideas.

Algorithms

Talks:

4. The Pandora 2D clustering and topological association algorithms
5. The Pandora 3D track reconstruction algorithms
6. The Pandora vertex reconstruction algorithms
7. The Pandora shower reconstruction and event building algorithms

Exercises:

2. Setup Pandora standalone development environment and add a new algorithm
3. Add algorithm implementation to perform 2D clustering operations
4. Add a new algorithm to perform 2D cluster merging, evaluate algorithm performance
5. Add a new algorithm to match 2D clusters between readout planes and form particles
7. (More complex) Add a new algorithm to perform particle merging operations
8. (More complex) Add a new algorithm to develop and apply track vs. shower selection cuts
9. (More wide-ranging) Work through Pandora ExampleContent learning library, then add new algs

Algorithms

- Don't yet have much of a feel for how long exercises will take; expect that they will take much longer than the place-holder slots listed on the agenda.
  - https://indico.fnal.gov/conferenceDisplay.py?confId=12043
- **Idea is not to stick rigidly to agenda, but to use any/all exercise time slots to work through the exercises at a pace comfortable/useful to you. Slower is probably better.**
- Fully expect to need time from days 3 and 4 to complete exercises. Completion of exercises means you are probably ready to contribute to Pandora LAr TPC reco
  - Essentially an infinite amount of work available! Would love to hear new ideas, etc.

The Key Exercises:

2. **Setup Pandora standalone development environment and add a new algorithm**
3. **Add algorithm implementation to perform 2D clustering operations**
4. **Add a new algorithm to perform 2D cluster merging, evaluate algorithm performance**
5. **Add a new algorithm to match 2D clusters between readout planes and form particles**

Questions?