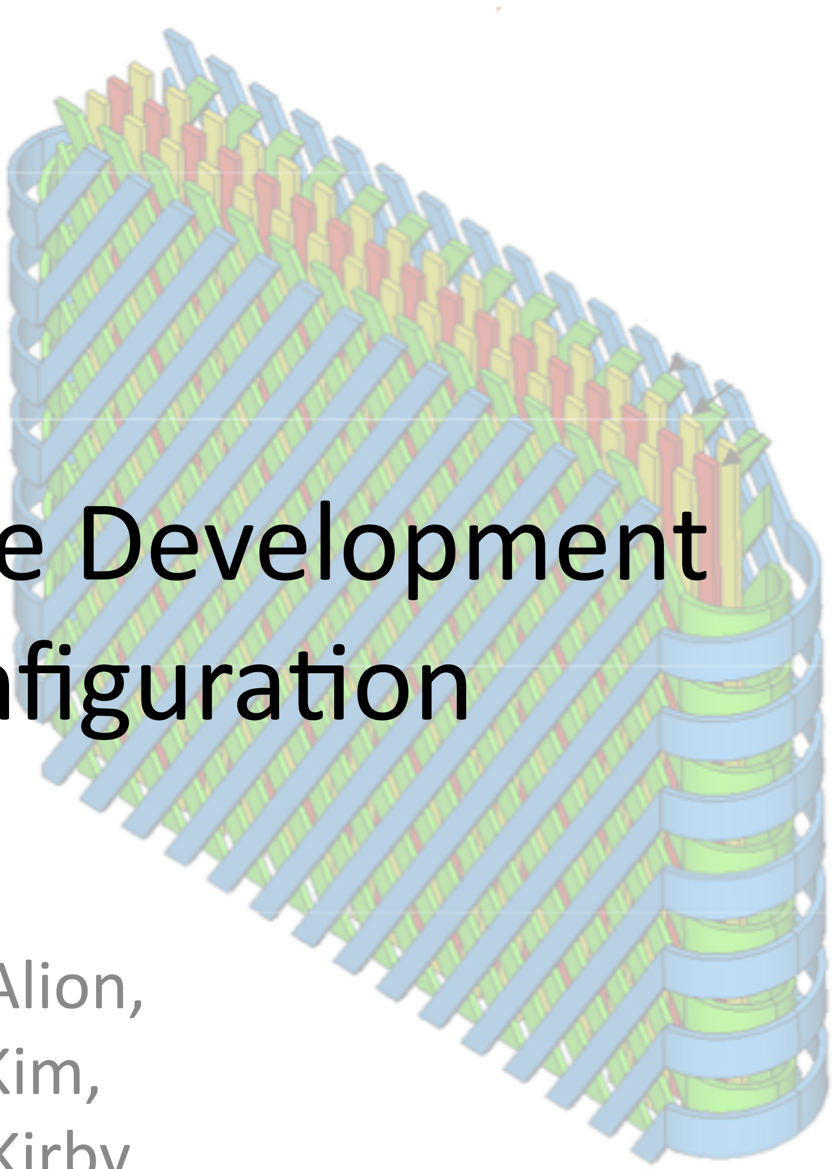
*No longer using std::multimap



Geometry Package Development for LBNE Configuration

Tyler Alion,
Jae Kim,
Mike Kirby,
Brian Rebel

ChannelToWire: Currently

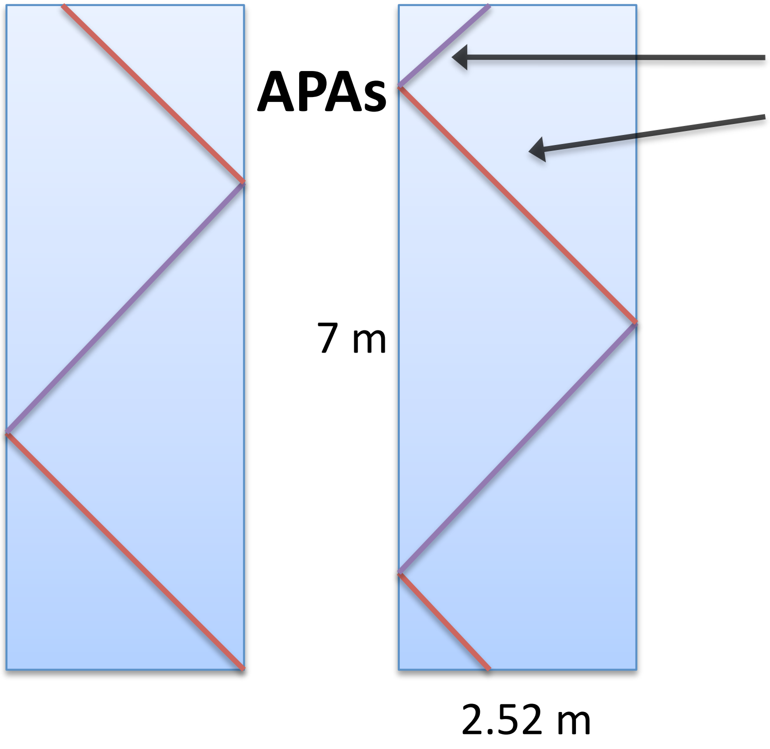
- Method
  - First call initialization to partition the channels among each plane in each TPC in each cryostat
  - Take input channel and find where it belongs using the data generated by the initialization
- Assumptions
  - Does not assume uniform number of wires per U plane / per V plane / per Z plane
  - While it handles multiple TPC's well, it is not aware of wire wrapping

ChannelToWire: Modification

- Structural changes:
  - Assume the same number of wires per each type of plane to speed it up a little
  - Loop over APA, where APA x is made of TPCs 2x and 2x+1;
    - the critical difference: channels exist in 2 TPCs but the same plane in every case, so looping over APA is better
    - Tricky to make this method also work for non-APA configurations like MicroBooNE – have several solutions, choosing among them.
- Changes that must be passed by the stakeholders:
  - Used to return a reference to a WireGeo obgect.
  - Now returns a vector of a struct that uniquely identifies each wire, called WireID

Input: Channel Number
Output: vector of WireID

Readout Channel on outer end (on top if the top APAs, bottom if the bottom APAs)



One readout channel can almost be thought of as one real life wire; to the code they are equivalent concepts.

Purple segments are behind
Red segments are on top

U,V: 3 or 4 code wire segments per channel
Vertical: 1 code wire segment per channel

```
struct WireID {unsigned int Cryostat,
               unsigned int TPC,
               unsigned int Plane,
               unsigned int Wire}
```

Return: Vector of WireID with size equal to the number of code wire segments contained by a channel

Other New Functions

- A reference to a WireGeo object is no longer returned
  - Require some change throughout all of the code
- Requires a new function which takes a vector of WireIDs and returns the WireGeo object reference(s)
  - Very simple, just better to separate the tasks
  - Maybe call it WireIDToWireGeo
- Also a function that takes channel, cryostat, TPC, and plane number
  - returns True if the given channel is in that cryostat,plane,TPC and returns False if not
  - Call it ChannelConsistency
  - Useful for limiting a loop to a specific

ChannelToWire: Example Use

```
std::vector<WireID> AllWires;
AllWires = ChannelToWire(chan);
wire = AllWires[i].Wire;
…
```

Looping through i will give you all the wires that need to be considered when looking at a channel

Or if you want to work with the object in order to use the other functions as well (More likely I guess)

```
std::vector<WireID> AllWires;
AllWires = ChannelToWire(chan);
this→WireIDToWireGeo(AllWires);
…
```

Schedule

- Last few steps to ensure compatibility with all configurations for each new method
- Modify code to be compatible with changes (Brian or Mike)
- Commit changes to be used!

- Then develop GDML while coordinating EventDisplay work

Questions?

Requests?

Suggestions?

Discussion?

The following slides are extra

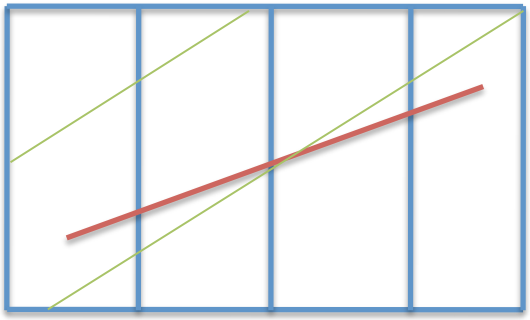
Some thoughts on EventDisplay

- Currently handles only one drift volume
  - With 3 sets of back to back drift volume pairs, will we display events that span multiple drift volumes
  - Assumed Yes – Feasible in 3D; but HOW in 2D?
- Currently handles only one TPC per drift volume
  - Event display must be aware of which TPC's respond to which drift volumes.
  - 3D very feasible; 2D needs an independent variable!

2D EventDisplay Independent Variable

- What can we choose X to be so that drift time's dependence on X give plenty insight into the event?
  - Used to be wire number (= channel number)
  - Wire# no longer makes sense with many TPCs
  - Channel# would give no intuitively useful plot since it is wrapped and doubled on each plane
  - Could create an independent variable mirroring the idea behind single-TPC event display

Top half of APA Arrangement →



The green line is a single TPC idea, where only one set of wires making a drift plane, makes sense as indep. variable.

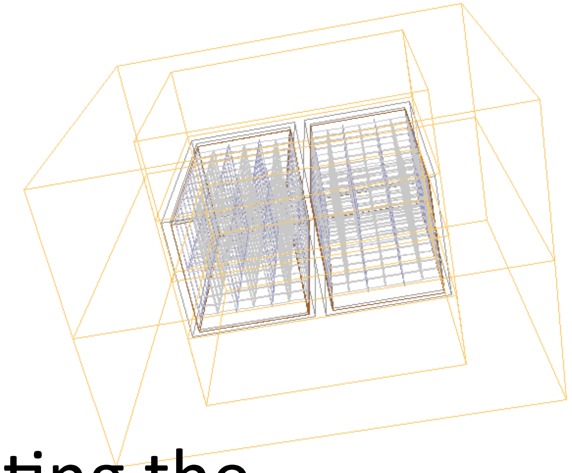
For this same idea to work with multi TPC, we would have to geometrically match up wires from bordering APAs.

GDML status

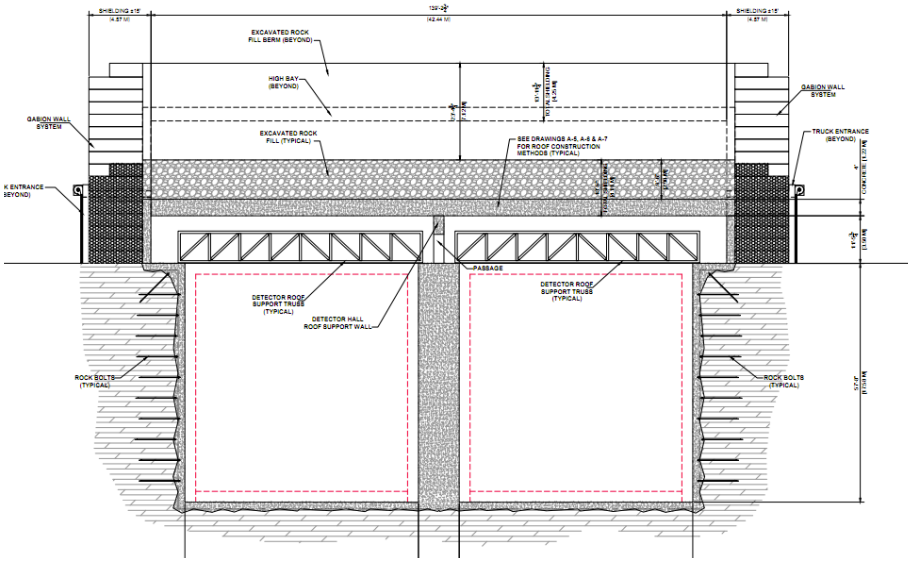
- Can generate events inside, but it is useless without the wire mapping to accommodate simulation.
- Need to add
  - field cage
  - highbay
  - some of the surrounding rock
- Are APA parameters up for debate again? It shouldn't be a problem, the generate script is mostly flexible to those parameters.

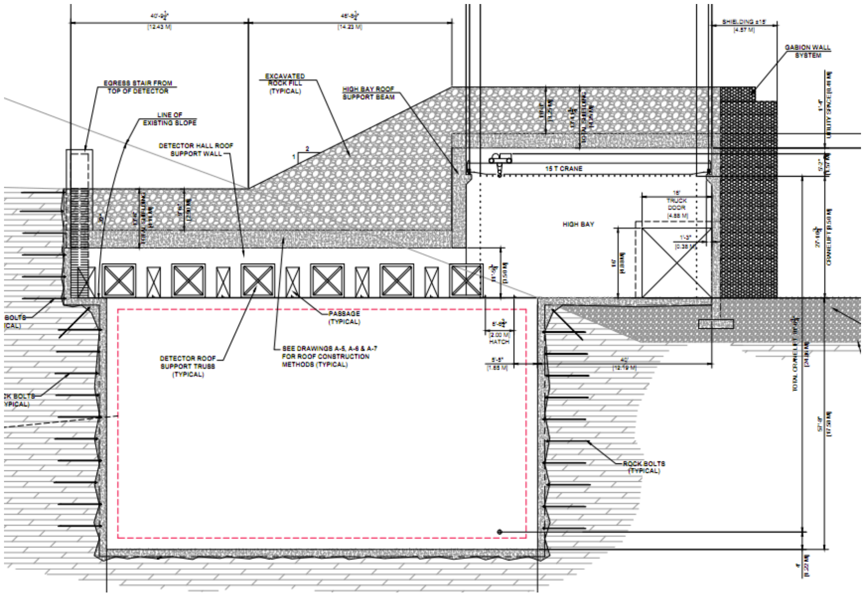
Still Underground

Detector Enclosure encased in DUSEL rock with thickness called RockThickness →

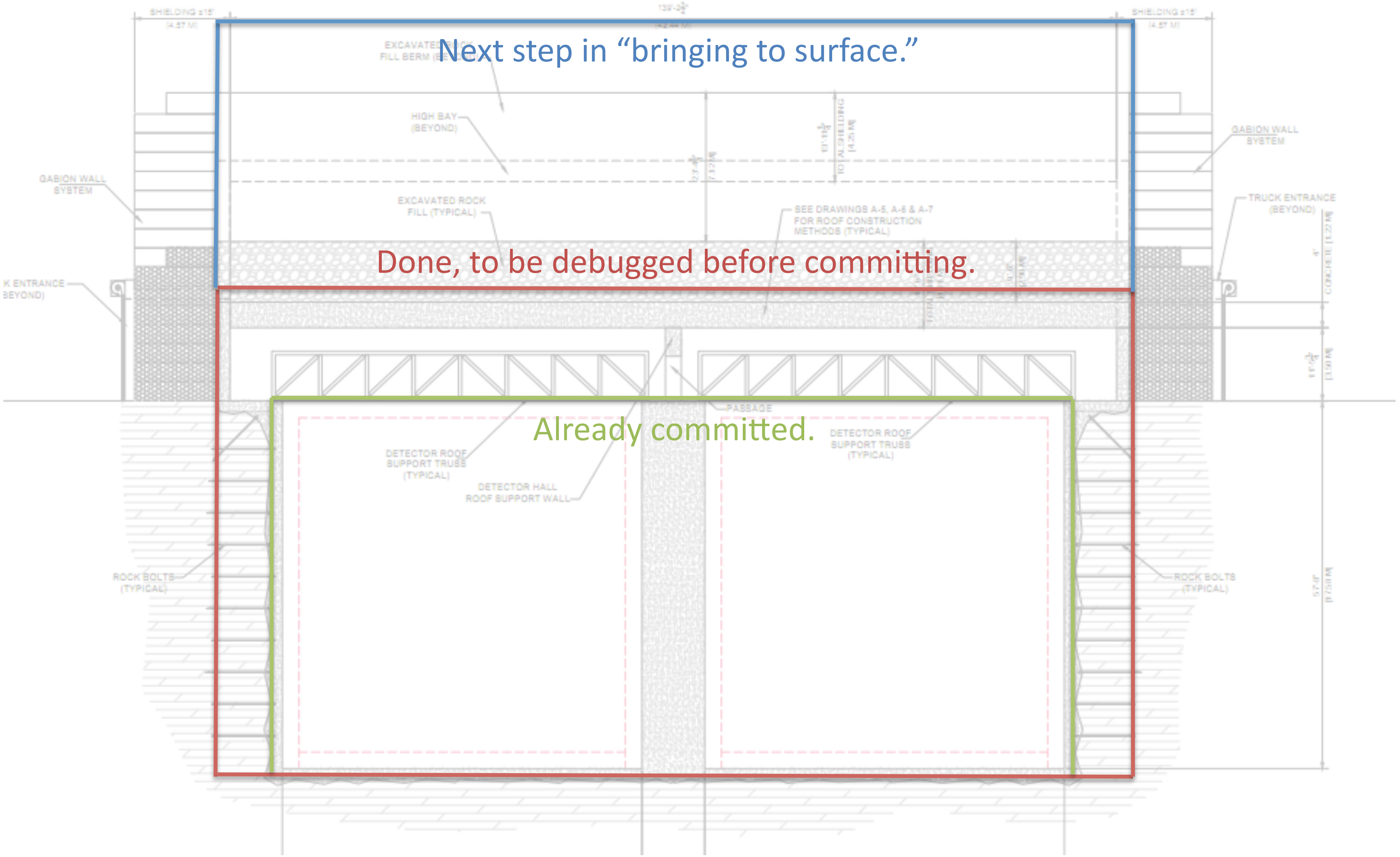


I have made a description representing the surface, currently debugging. ~1-2 weeks to go.





Once debugged, on to the highbay.

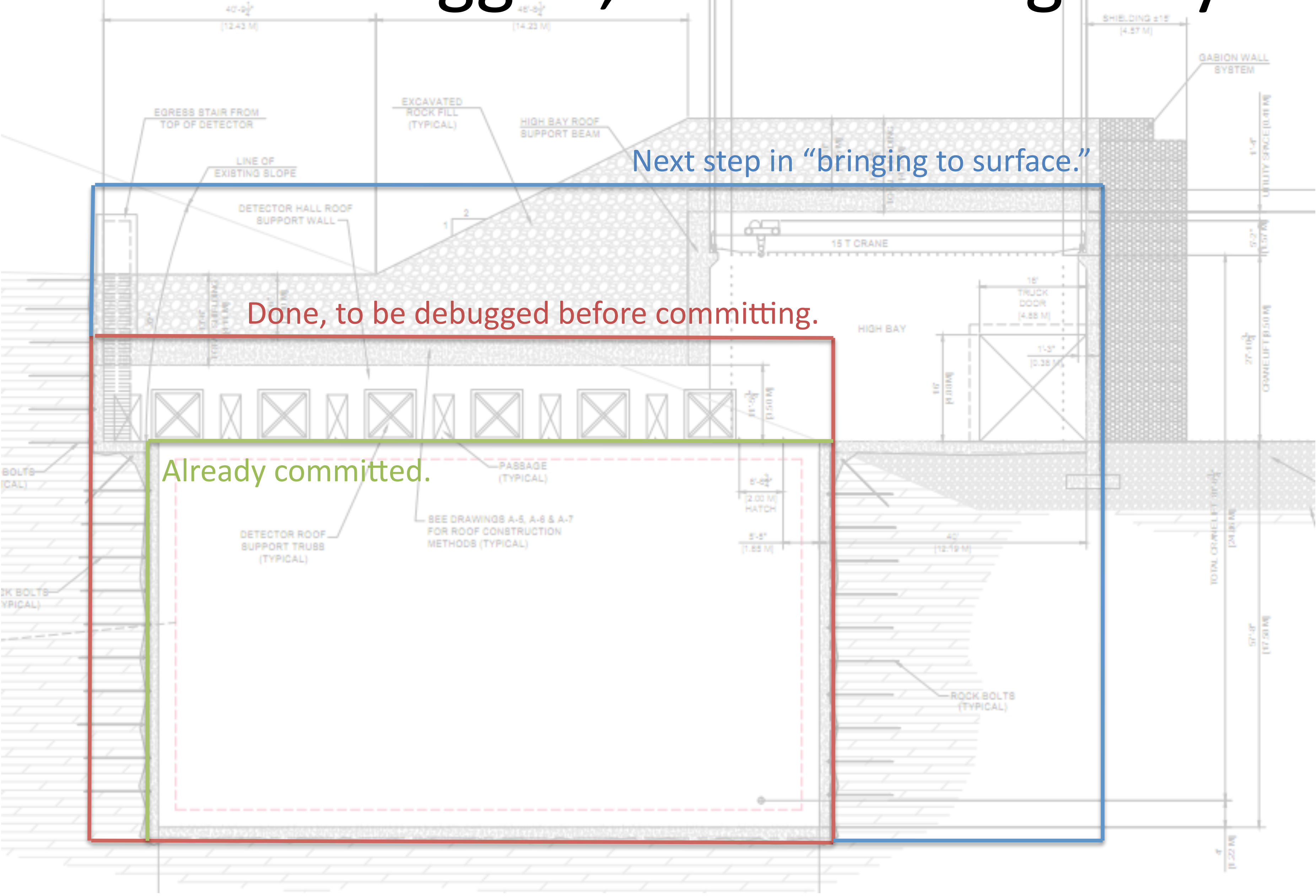


Next step in "bringing to surface."

Done, to be debugged before committing.

Already committed.

Once debugged, on to the highbay.

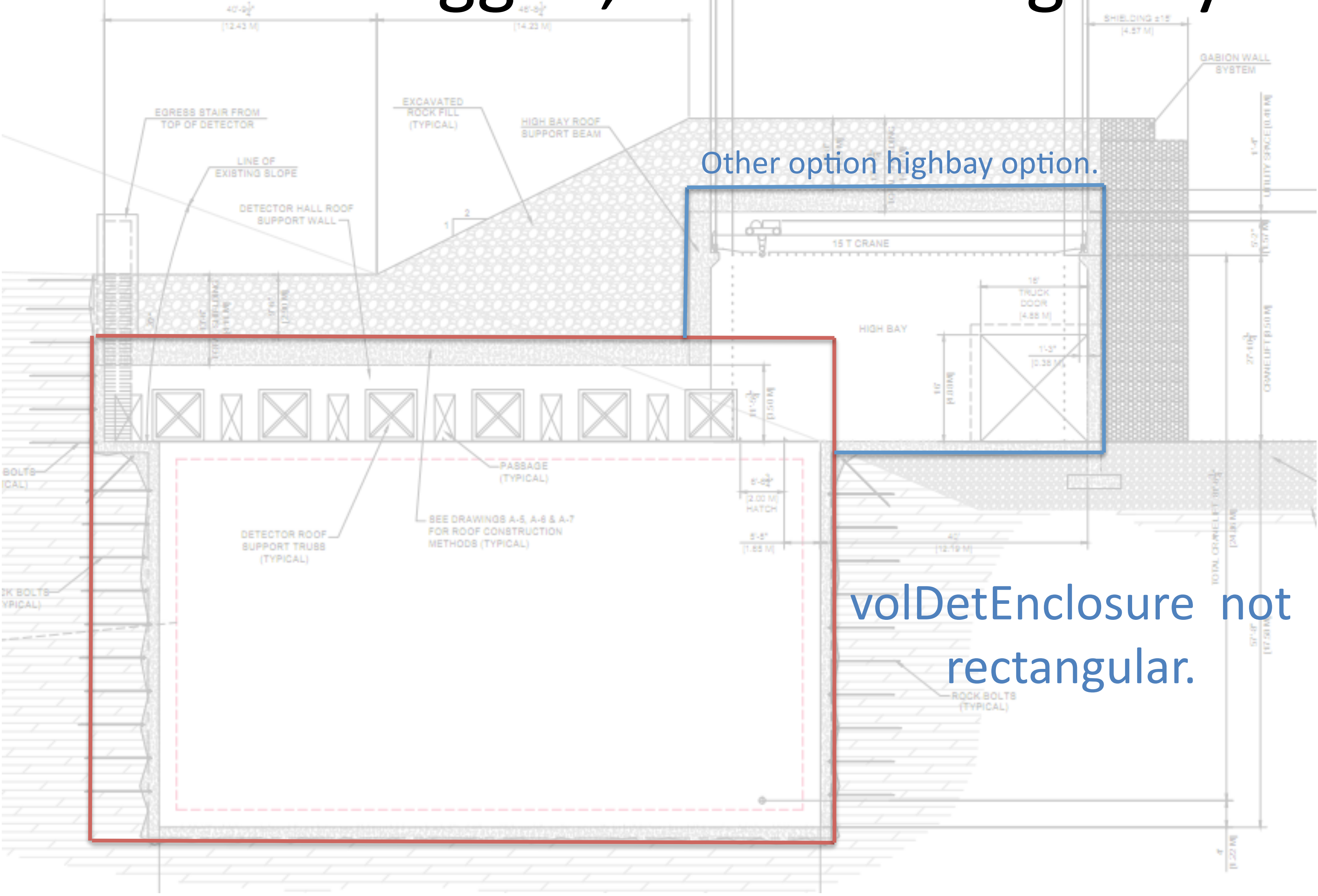


Next step in “bringing to surface.”

Done, to be debugged before committing.

Already committed.

Once debugged, on to the highbay.



Other option highbay option.

volDetEnclosure not rectangular.