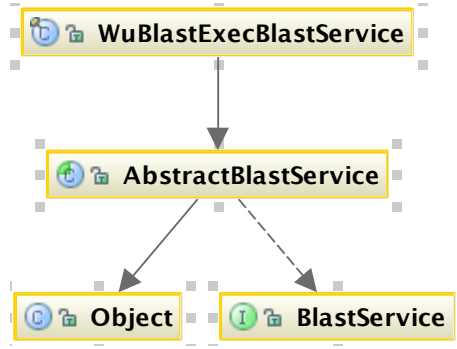
Putting:

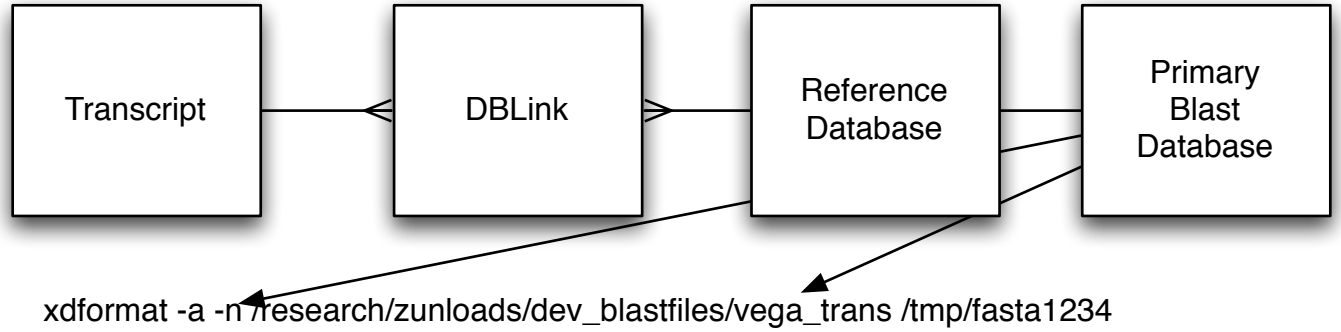


- All blast access occurs through BlastService interface.
  - singleton access

```
Sequence sequence =
WuBlastExecBlastService.createWuBlastExecBlas
tService().addSequenceToTranscript(transcript
DTO.getZdbID(),sequenceDTO.getData(),referenc
eDatabaseDTO.getZdbID()) ;
```

- **Sequence** is a presentation object wrapping a blast sequence with a reference to its DBLink

- Create dblink and accession bank entries from autogenerated accessions.
  - BlastService.addSequenceToTranscript -> WuBlastExec.addSequence
  * BlastService.addProteinToMarker -> WuBlastExec.addSequence
- Validate sequence
- Create FASTA file with cool defline and write to temp.
- Use xdformat to create entry.



xdformat -a -n /research/zunloads/dev_blastfiles/vega_trans /tmp/fasta1234

```
        File fastFilePath = File.createTempFile("fasta","fa");
            BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(fastFilePath)) ;
            bufferedWriter.write(sequence.getFormattedData());
            bufferedWriter.close();
            List<String> commandList = new ArrayList<String>();
            commandList.add("/private/apps/wublast/xdformat");
            commandList.add("-"+type);
            commandList.add("-a");
            commandList.add(blastDBPath);  // vega transcript)
            commandList.add(fastFilePath.getAbsolutePath());


        Process process = Runtime.getRuntime().exec(commands);
//              todo: add buffer output onto process . .. may need to
use threading
            BufferedReader stderr = new BufferedReader(new
InputStreamReader(process.getErrorStream()));
            BufferedReader stdout = new BufferedReader(new
InputStreamReader(process.getInputStream()));
```