# Software Architecture

Software engineering patterns
MVC, Command, Configuration, Singleton
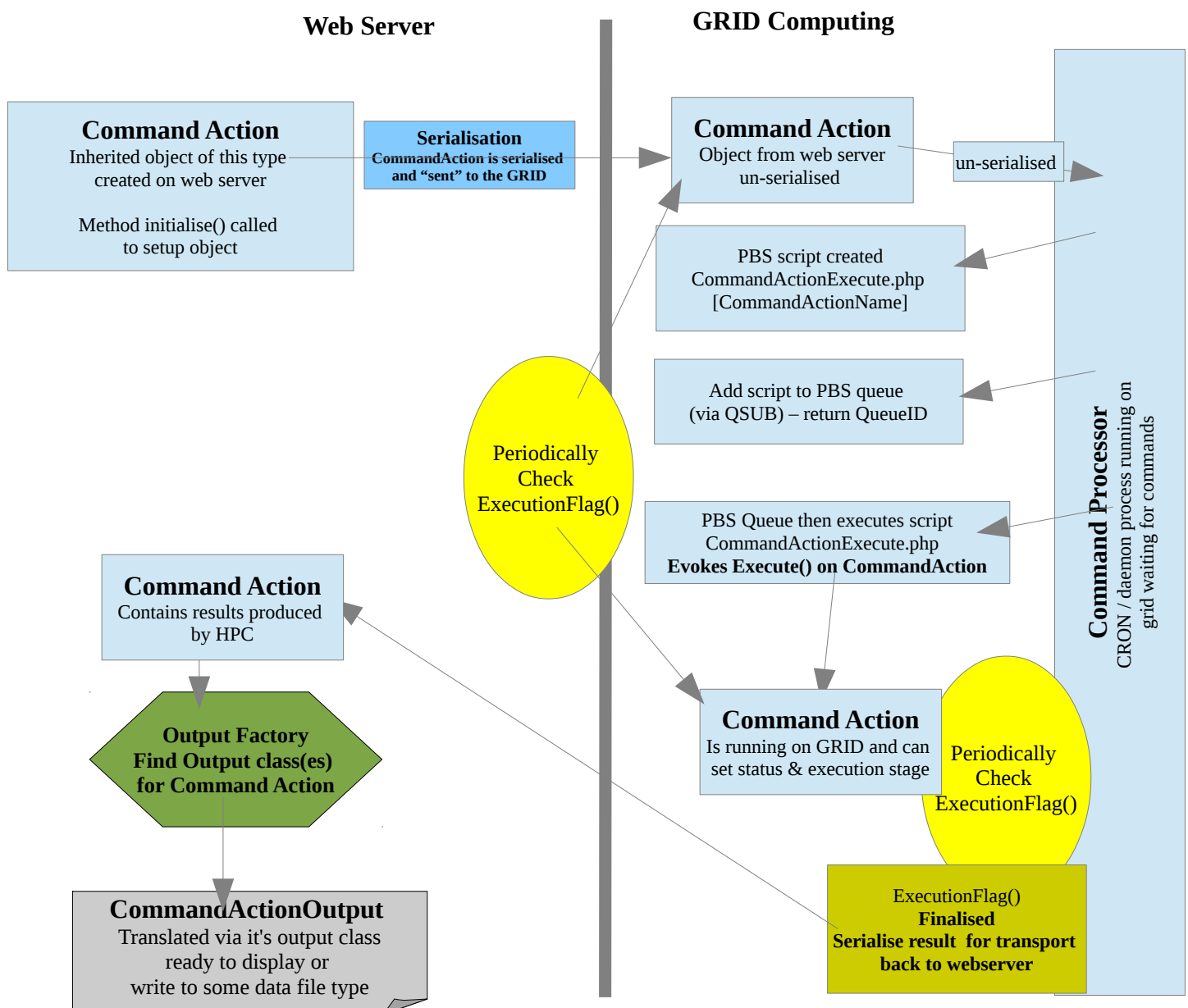
- – Remote Job execution (Command Pattern)
- – Spatial File conversion (MapServer wrapper)
- – User Input Selection, Output element Display

**Remote Job execution** *(Command Pattern)*
Requirement: To execute "PBS" jobs in the JCU (or other) Grid Computing Environment (HPC)

Why not just use the current QSUB and QSTAT with some TCSH scripts?
*For the AP02 project, we need the ability to send complex commands to the HPC and receive complex messages / results.*

**Web Server** | **GRID Computing**

**Command Action**
Inherited object of this type created on web server

Method initialise() called to setup object

**Serialisation**
*CommandAction is serialised and "sent" to the GRID*

**Command Action**
Object from web server un-serialised

un-serialised

PBS script created
CommandActionExecute.php
[CommandActionName]

Add script to PBS queue
(via QSUB) – return QueueID

Periodically Check ExecutionFlag()

PBS Queue then executes script
CommandActionExecute.php
**Evokes Execute() on CommandAction**

**Command Action**
Contains results produced by HPC

**Output Factory
Find Output class(es)
for Command Action**

**CommandActionOutput**
Translated via it's output class ready to display or write to some data file type

**Command Action**
Is running on GRID and can set status & execution stage

Periodically Check ExecutionFlag()

ExecutionFlag()
**Finalised
Serialise result for transport back to webserver**

**Command Processor**
CRON / daemon process running on grid waiting for commands

## CRON job

A standard unix "cron" job, to execute Incoming.php (currently every minute)
- this job can also be run me a process that can watch for chnages in a folder


## Process Queue

get list of command files in Queue folder (.command)
(a command file is a serialised version of a CommandAction)

### Process Single Queue Item
Read in command file, unserailize via ( CommandUtil::GetCommandFromFile() )
check class type, is CommandActionm, or sub class of then process


Process types based on value of ExecutionFlag – an execution  state.

**READY**   awaiting to be started, create scripts suitable for PBS,  add scripts into PBS queue (QSUB)
       set to RUNNING

**RUNNING**  Go and check QSUB status , if job can't be found then FINALISE,
       as the job itself did signal; that it has finalised

**QUEUE_DONE**Job has completed as far as the QSTAT is concerned, though the command still reads RUNNING.
       This is OK, as a job does not have update it's on execution state, though if it does it's can send more
       meaningful status back to the calling system

**FINALISE**  Has finished - but not yet complete, is set by the process updating it's own execution  state.


**COMPLETE** Has completed - results to be returned collect results of CommandAction as send back
       to calling system, via the serialised object


## CommandAction Object Hierarchy

Object
   CommandAction
     SpeciesMaxent


### Files

| generic_job.sh | CRON job running every minute on HPC looking for new jobs in queue folder |
|---|---|
| Incoming.php | simple php script to include libraries & execute CommandProcessor |
| CommandProcessor.php | Watches queue folder of commands to be processed. |
| SpeciesMaxent.action.php | CommandAction wrapper to Maxent, |
| CommandAction.class.php | Super Class of any action to be run on grid |