aLIGO Guardian
Status Update

Jameson Graef Rollins

LIGO Caltech

LLO Commissioning workshop
June 12, 2014

Guardian user manual

This is a overview of the development and deployment status of the Advanced LIGO **Guardian** automation system.

For information on how Guardian works, see the Advanced LIGO Guardian Overview and User Manual (LIGO-G1400016).

Deployment status

Core code

Guardian is based on Python. System modules (user code), as well as the core execution code, are written in **python**.

Two core libraries:

guardian Core system graph and state execution libraries, helper utilities, command line interfaces, MEDM control interface, etc. (currently v1011 at LLO)

cdsutils Ezca CDS EPICS client library, other CDS interaction libraries and CLI (currently v238 at LLO)

The core code is now quite mature and (fairly) stable. It is installed at both sites and at development/testing facilities at Stanford and MIT LASTI.

Site infrastructure

Node supervision infrastructure has been fully built out at both sites, running on dedicated machines in the CDS front-end networks ({h1,l1}guardian0).

node	s	m	n	vers	state	message
HPI_BS	*	M	!	1011	READY	Some filters are not in the expected sta
HPI_ETMX	*	M	!	1011	READY	Some filters are not in the expected sta
HPI_ETMY	*	M	-	1011	READY	-
HPI_ITMX	*	M	!	1011	READY	Some filters are not in the expected sta
HPI_ITMY	*	M	!	1011	READY	Some filters are not in the expected sta
IFO_IMC	*	E	-	1011	LOCKED	-
IFO_LOCK	*	E	-	1011	IDLE	-
ISI_BS_ST1	*	M	-	1011	HIGH_ISOLATED	-
ISI_BS_ST2	*	M	-	1011	DAMPED	-
ISI_ETMX_ST1	*	M	-	1011	HIGH_ISOLATED	-
ISI_ETMX_ST2	*	M	-	1011	DAMPED	-
ISI_ETMY_ST1	*	M	-	1011	HIGH_ISOLATED	-
ISI_ETMY_ST2	*	M	-	1011	DAMPED	-
ISI_HAM2	*	E	-	1011	HIGH_ISOLATED	-
ISI_HAM3	*	E	-	1011	MEDIUM_ISOLATED	-
ISI_HAM4	*	E	-	1011	MEDIUM_ISOLATED	-
ISI_HAM5	*	E	-	1011	MEDIUM_ISOLATED	-
ISI_HAM6	*	E	-	1011	MEDIUM_ISOLATED	-
ISI_ITMX_ST1	*	M	-	1011	HIGH_ISOLATED	-
ISI_ITMX_ST2	*	M	-	1011	DAMPED	-
ISI_ITMY_ST1	*	M	-	1011	HIGH_ISOLATED	-
ISI_ITMY_ST2	*	M	-	1011	DAMPED	-
SEI_BS	*	E	-	1011	ISOLATED_DAMPED	-
SEI_ETMX	*	E	-	1011	ISOLATED_DAMPED	-

User code status

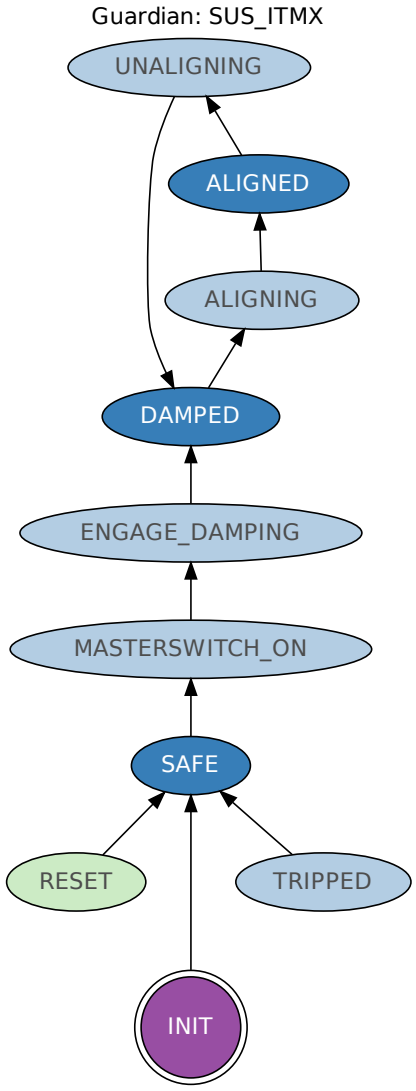
System modules, i.e. user code, i.e. *actually automation logic*, under heavy developed, and progressing nicely.

Commissioned nodes

SUS was the first subsystem automated (also the simplest). All suspensions nodes have identical guardian structure, replying upon a base SUS.py module and a sustools.py library interface.

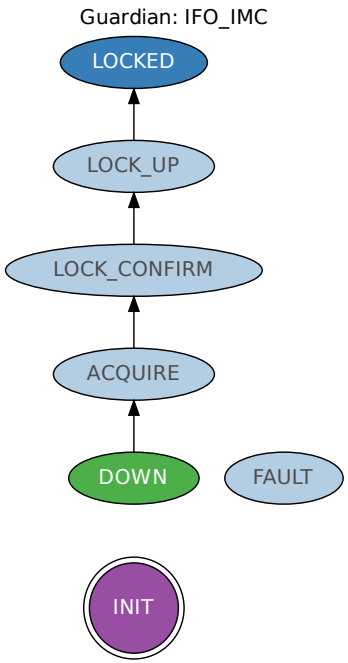
Fully handles recovery to aligned state from watchdog trips.

Currently (as of this week at LLO) *not* touching the alignment offsets. Will need to revisit how alignment offsets are managed.

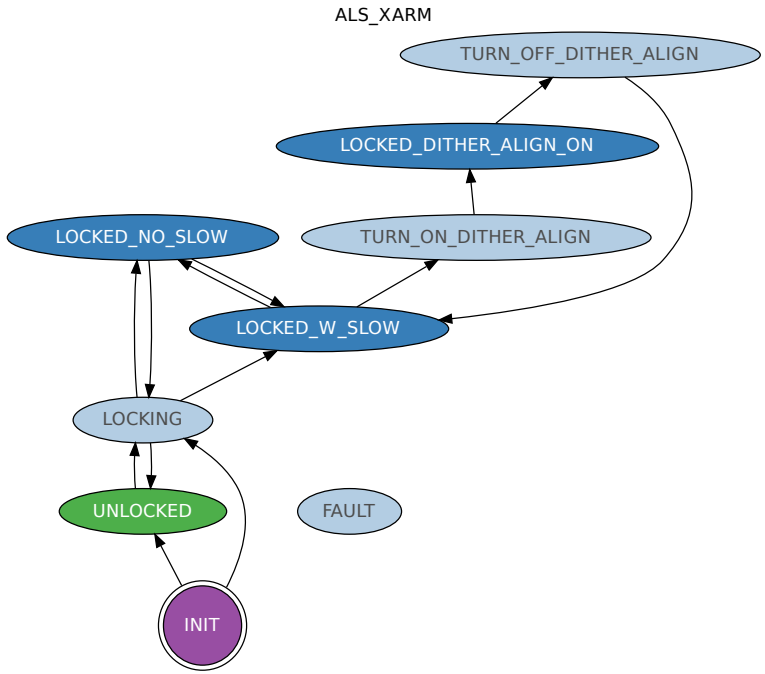


Commissioned nodes

The **IMC** was next, and was the first "non-trivial" control. It is now deployed at both sites (although still being tweaked).



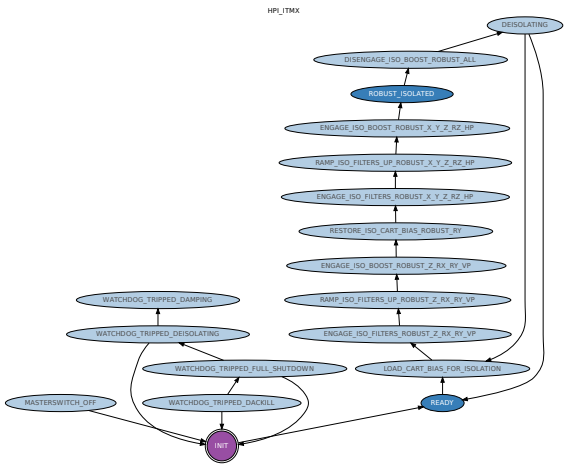
The **ALS** system under active development at LHO.



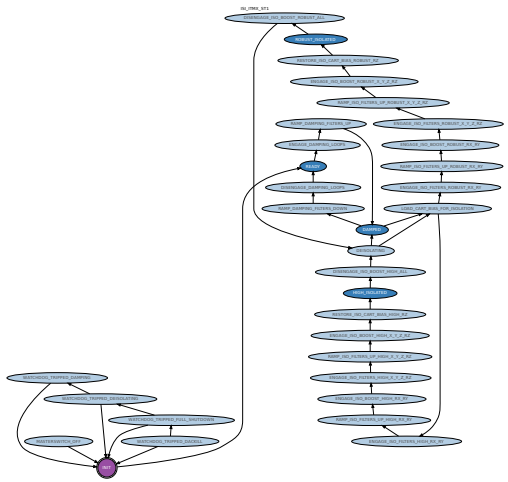
Commissioned nodes

Next was **SEI** at LHO, by far the most complicated system currently deployed. Fully handles recovery to aligned state from watchdog trips. Chamber seismic system consists of multiple nodes managed by “chamber manager” (HAM: 3 nodes, BSC: 4 nodes):

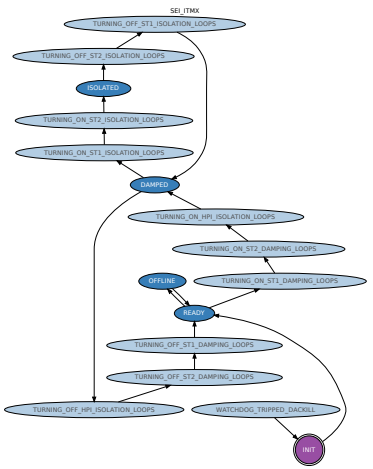
HPI_ITMX



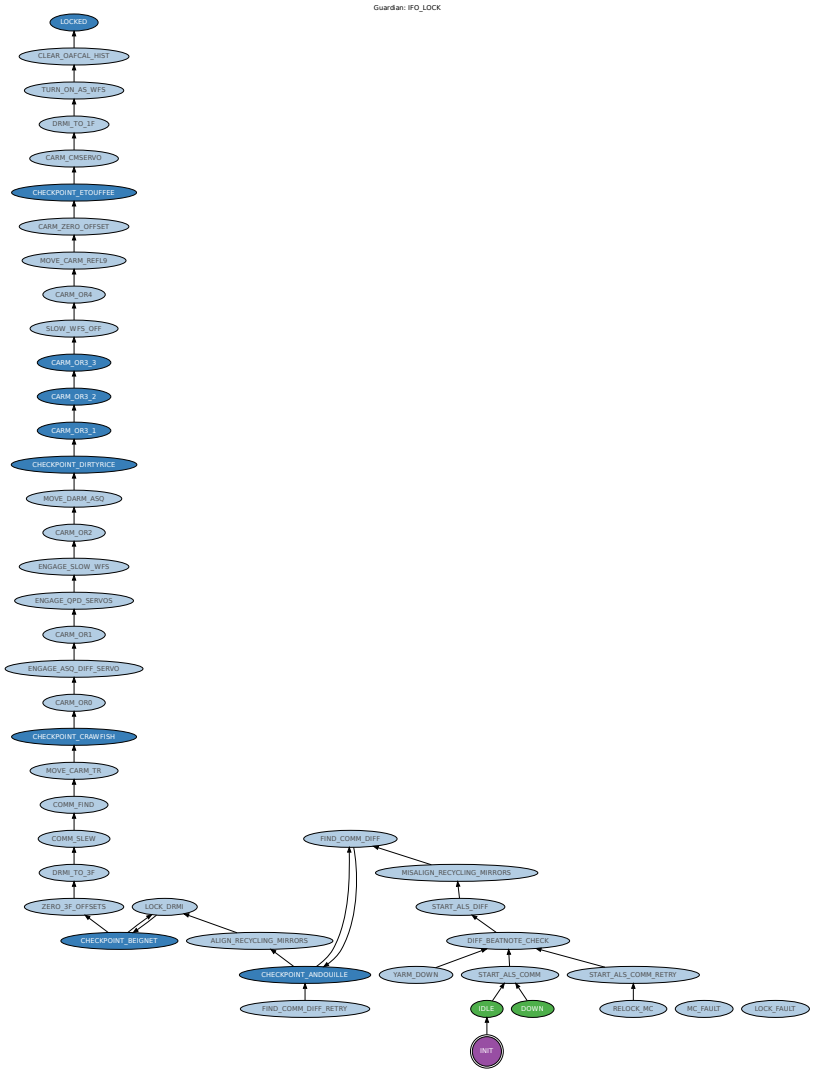
ISI_ITMX_ST{1,2}



SEI_ITMX (manager)



Commissioned nodes



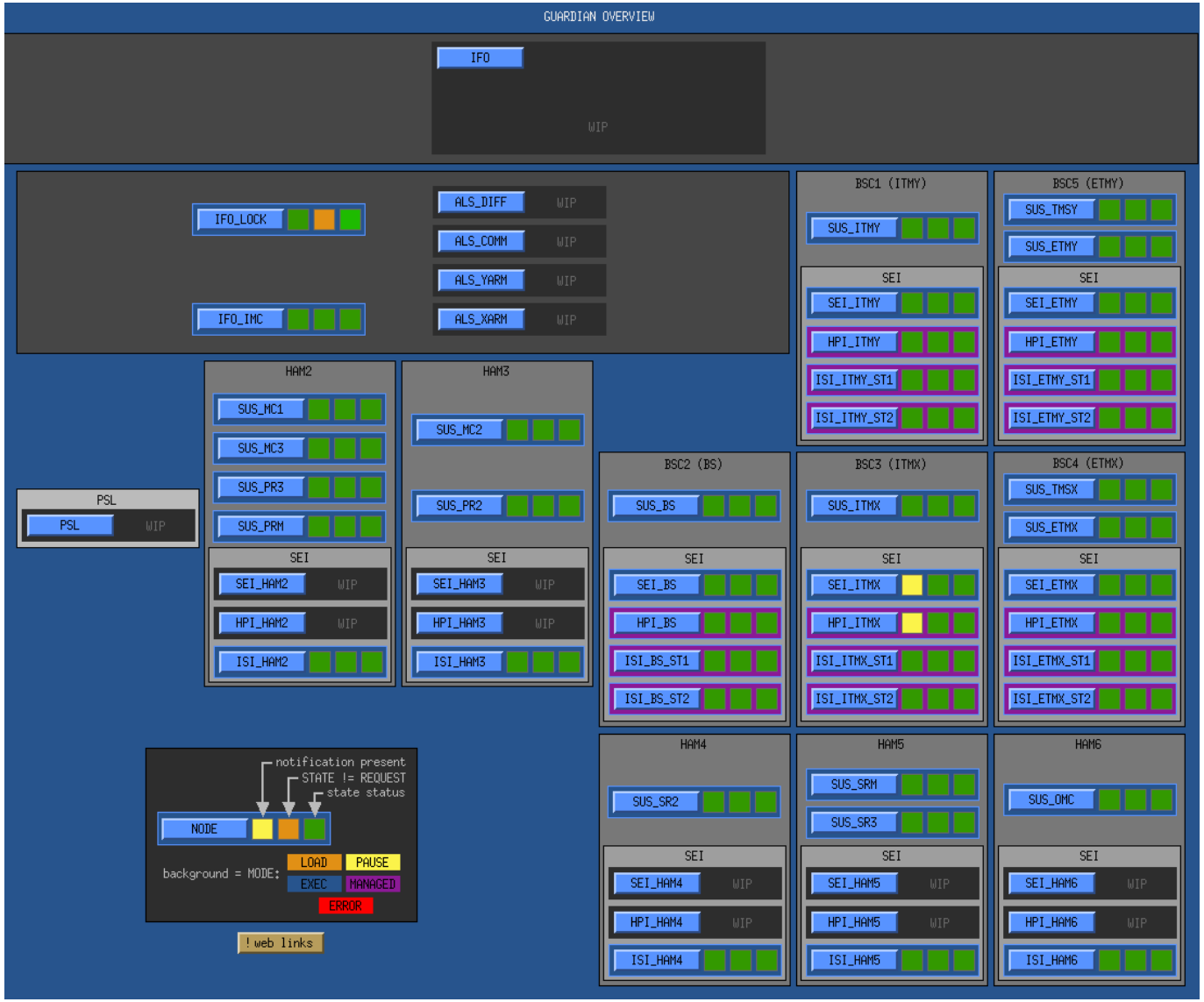
Hot new node: **IFO_LOCK** full IFO lock acquisition node at LLO!

This node is also under heavy development, but starts to give a sense of what the full lock acquisition procedure will look like.

User code status

- SUS deployed at both sites.
- IMC deployed at both sites.
- SEI deployed at both sites.
- ISC is now primary focus, heavy development at LLO this summer.
- NodeManager infrastructure under development and testing. First heavily used in SEI, and will be very important for the rest of the IFO locking.
- Still needed: TCS, PSL, others?

Full system overview



Full system overview

At this point it's **very important** to pay attention to this GUARD_OVERVIEW screen. It tells us the status of the guardian processes that are *actively controlling the instrument* (e.g. which nodes are running, whether there are errors, notifications, etc.).

The **orange** lights indicate that the node STATE does not equal the REQUEST, i.e. the node is actively trying to achieve the request.

The **yellow** lights indicate a NOTIFICATION is present, meaning something requires operator intervention. For instance, this will tell us if any of the SUS or SEI systems has a watchdog trip.

Full system overview