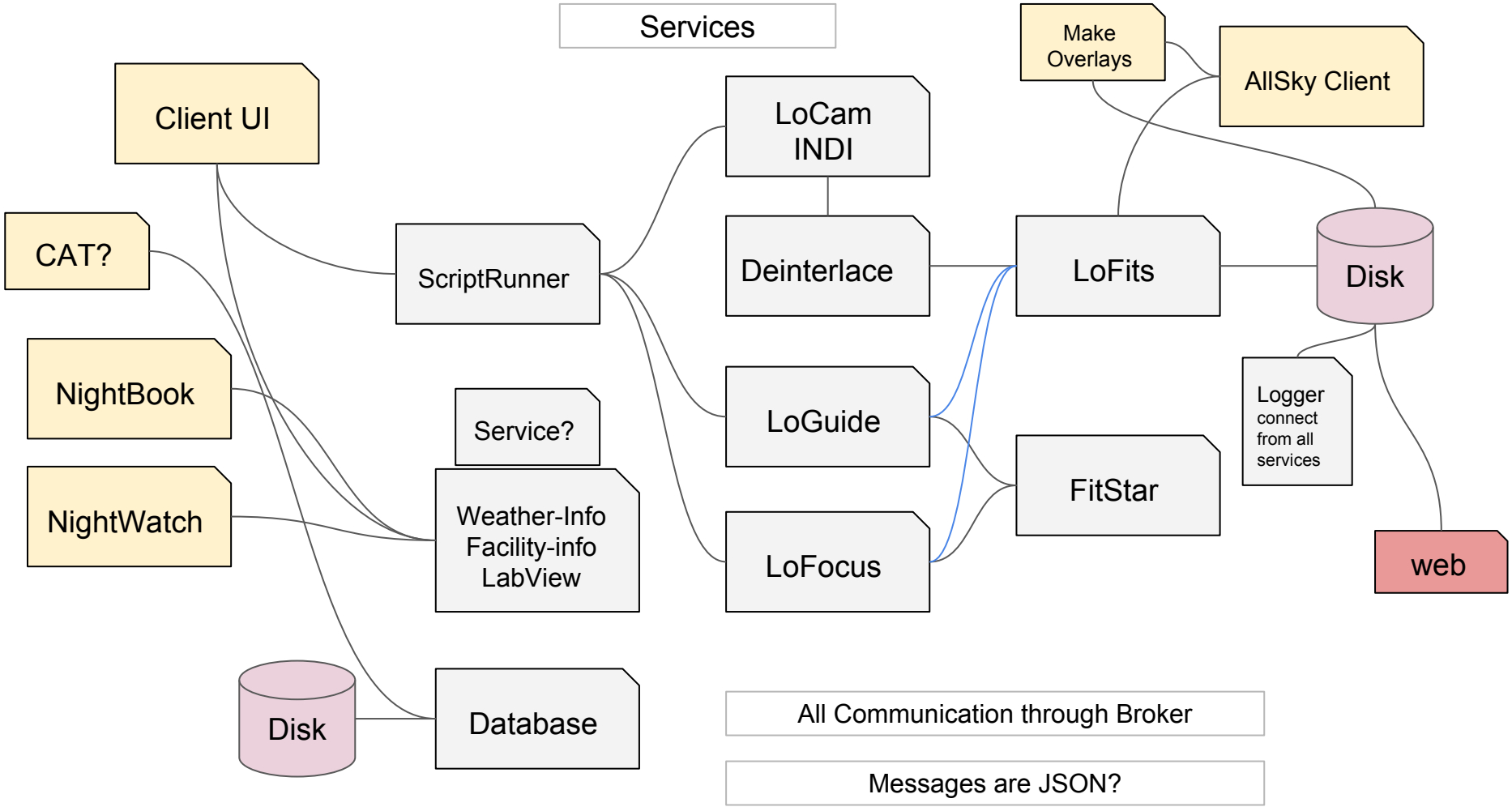
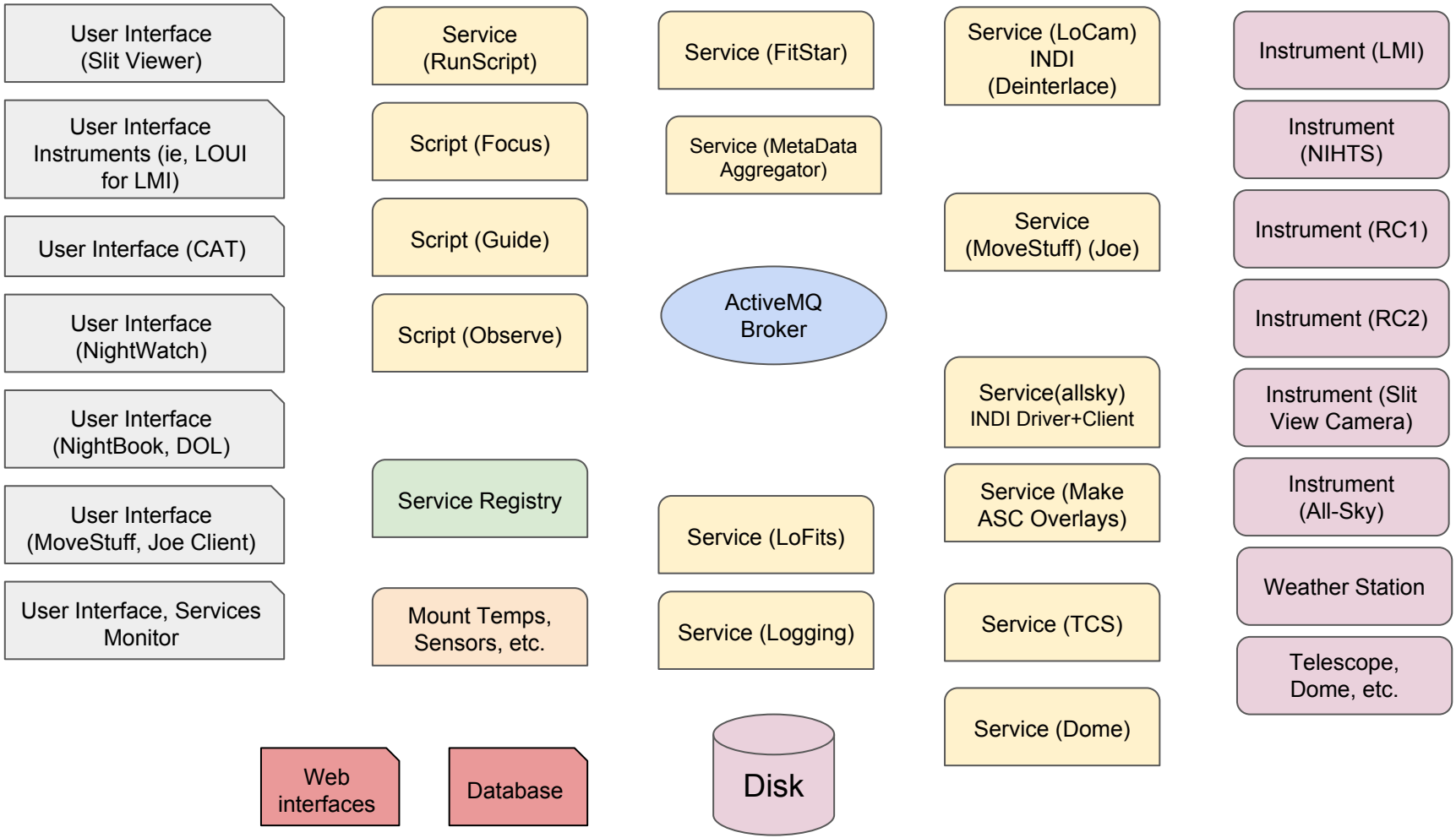
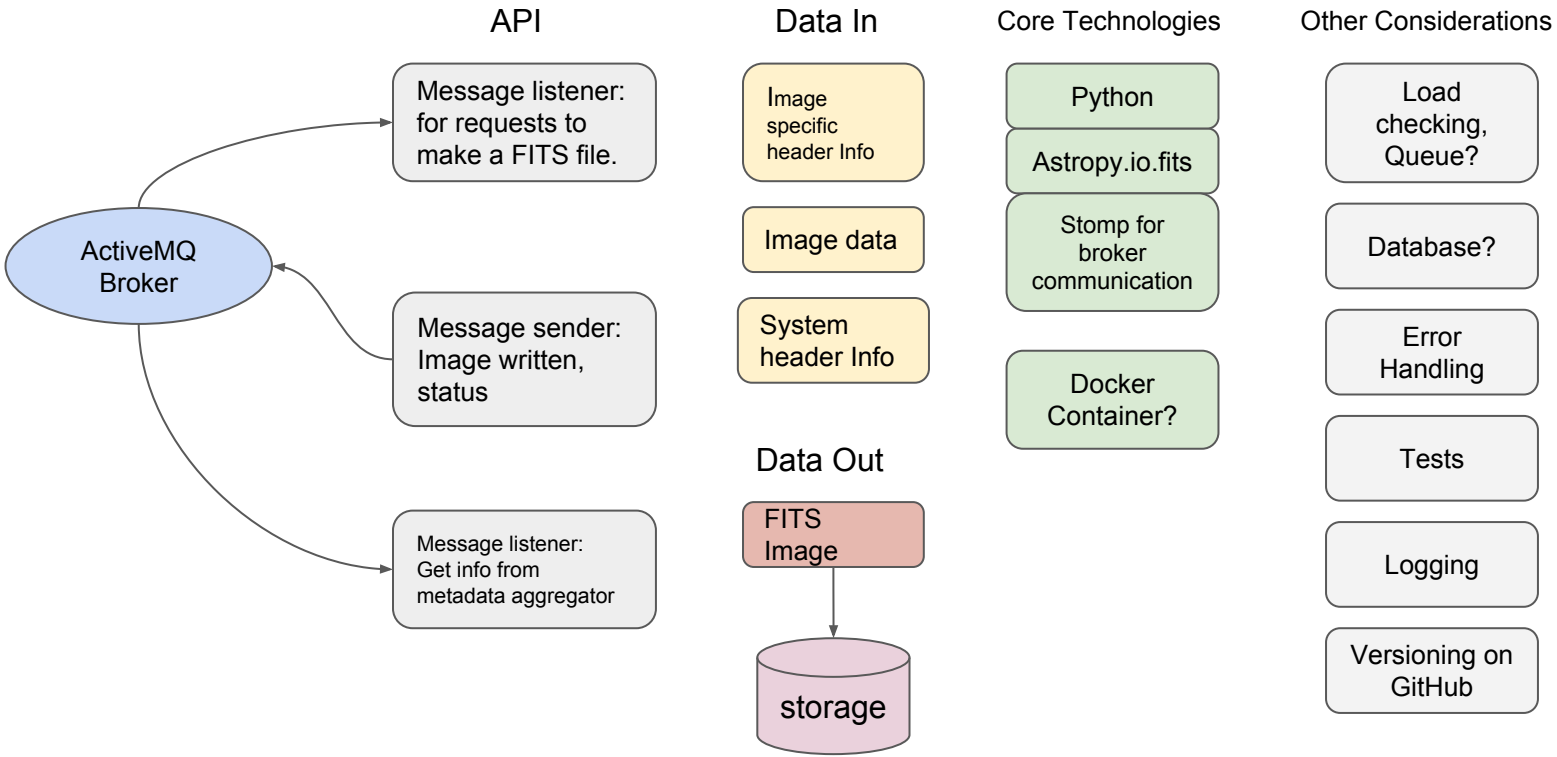
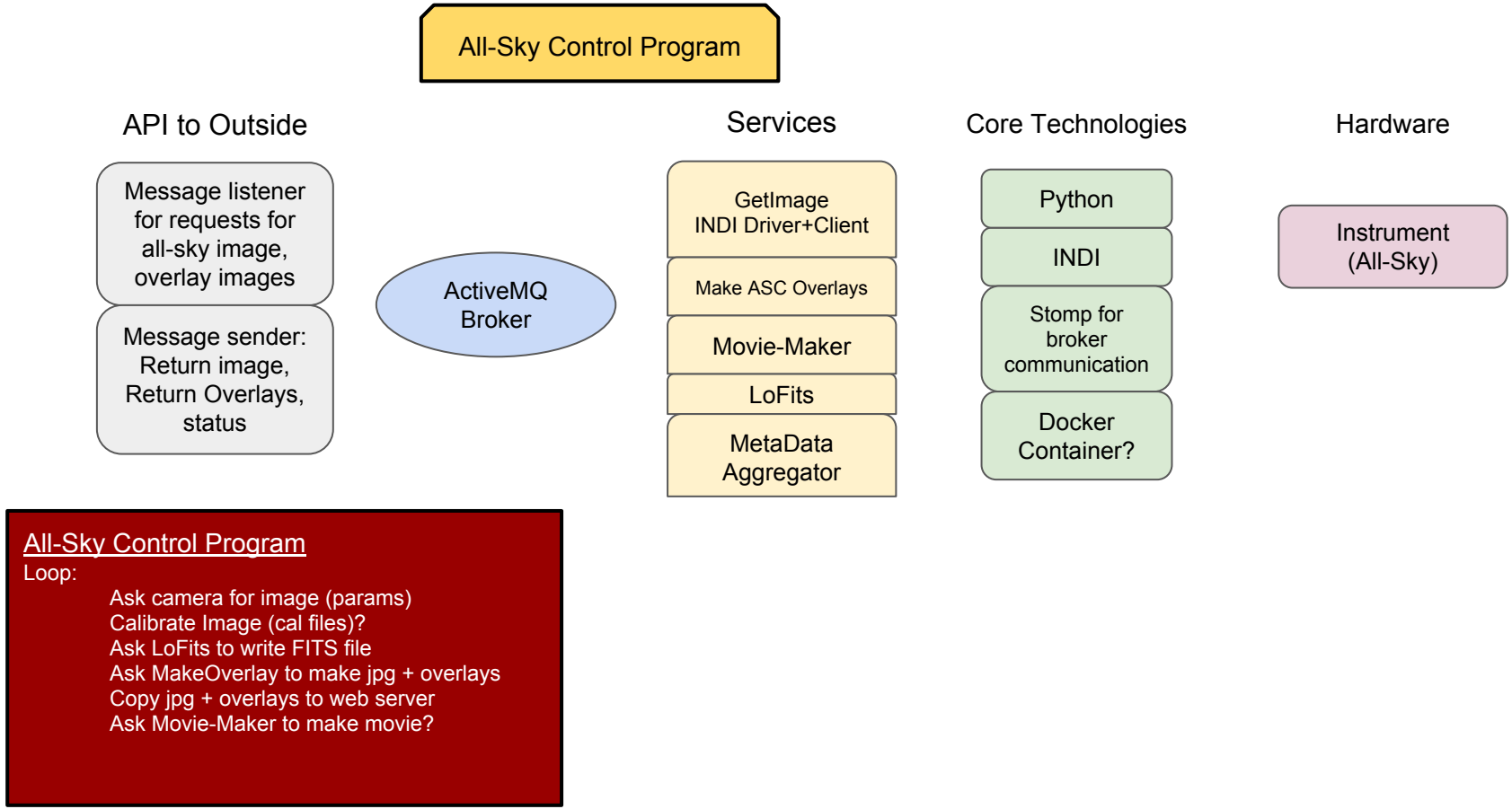

Services
Client UI
CAT?
NightBook
NightWatch
ScriptRunner
Service?
Weather-Info
Facility-info
LabView
Disk
Database
LoCam
INDI
Deinterlace
LoGuide
LoFocus
Make
Overlays
AllSky Client
LoFits
FitStar
Disk
Logger
connect
from all
services
web
All Communication through Broker
Messages are JSON?


User Interface (Slit Viewer)
User Interface Instruments (ie, LOUI for LMI)
User Interface (CAT)
User Interface (NightWatch)
User Interface (NightBook, DOL)
User Interface (MoveStuff, Joe Client)
User Interface, Services Monitor
Service (RunScript)
Script (Focus)
Script (Guide)
Script (Observe)
Service Registry
Mount Temps, Sensors, etc.
Service (FitStar)
Service (MetaData Aggregator)
ActiveMQ Broker
Service (LoFits)
Service (Logging)
Service (LoCam) INDI (Deinterlace)
Service (MoveStuff) (Joe)
Service(allsky) INDI Driver+Client
Service (Make ASC Overlays)
Service (TCS)
Service (Dome)
Instrument (LMI)
Instrument (NIHTS)
Instrument (RC1)
Instrument (RC2)
Instrument (Slit View Camera)
Instrument (All-Sky)
Weather Station
Telescope, Dome, etc.
Web interfaces
Database
Disk

Architecture of LoFits Service



Architecture of All-Sky Software and Services



All-Sky Control Program

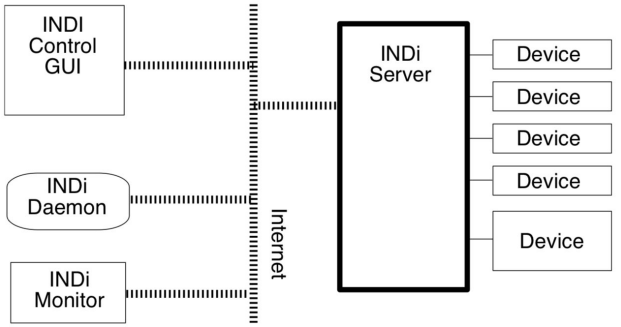
Loop:

- Ask camera for image (params)
- Calibrate Image (cal files)?
- Ask LoFits to write FITS file
- Ask MakeOverlay to make jpg + overlays
- Copy jpg + overlays to web server
- Ask Movie-Maker to make movie?

Architecture evolution at Lowell

Current architecture is client/server with other services mixed in.

- LOUI/LOIS is client/server and they are tightly coupled.
- The TCS/AOS/ECS/OCS/CCS/DCS/etc. System acts as a Service to CAT/LOUI but also has its own control panel.
- Architecture of new system decentralized and loosely coupled using asynchronous communication through broker.
- Loose coupling means that many different clients can access a particular service because it isn't tightly coupled to any one client.
- All components publish their services in a standardized, cross-platform manner.



INDI with an intermediate Server

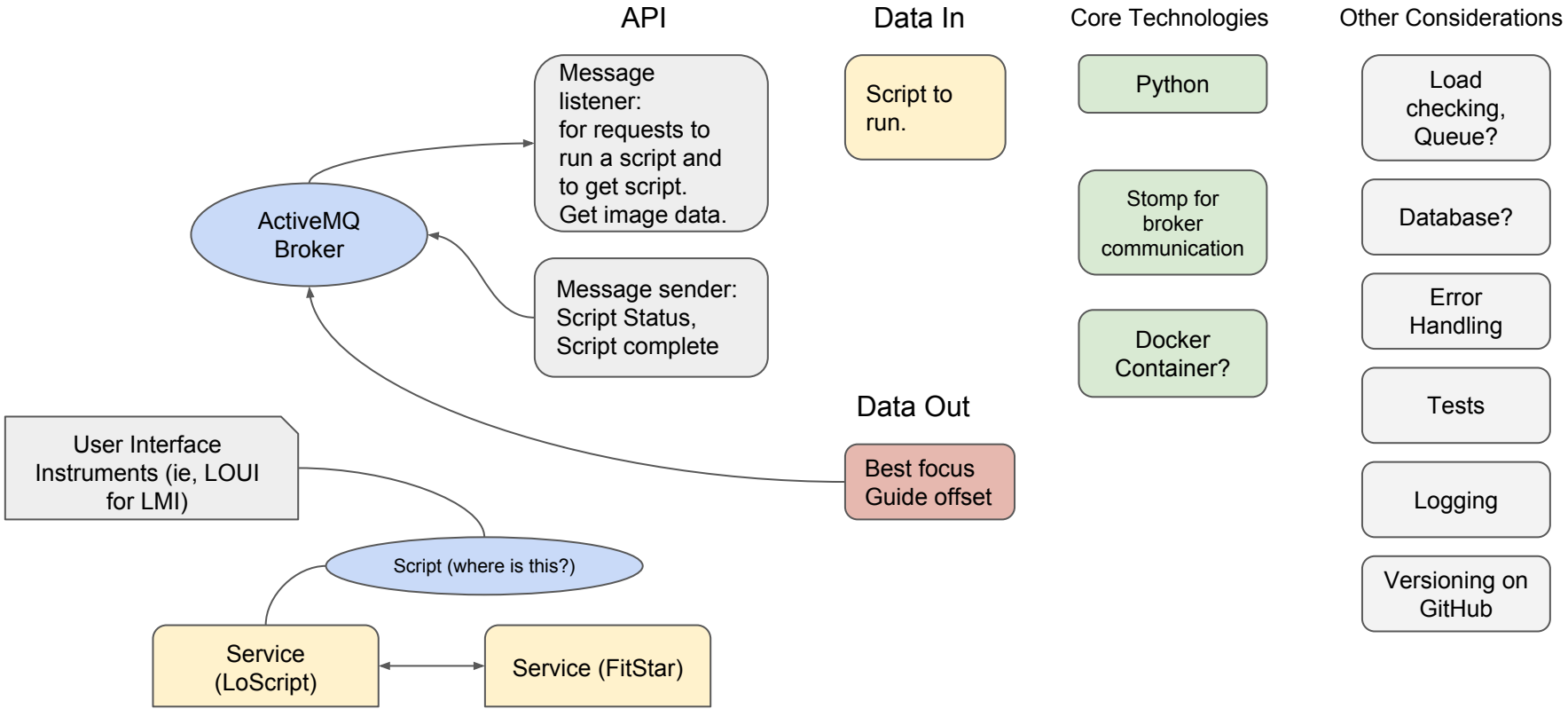
Communication Structure, ideas from the INDI protocol.

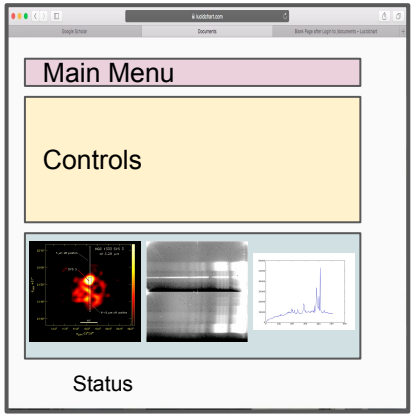
- An INDI device casts it services as a set of properties in the INDI format.
- An INDI client is a process which connects to an INDI device, queries the device for its set of control properties, and possibly sends requests to change those properties to the device.
- The protocol is designed to support arbitration and broadcasting among several clients and devices.
- **Text** properties, **number** properties, **switch** properties, **light** properties and **blob** properties
- Name, label, permission, state, timeout, etc.
- Each command between Client and Device specifies a **Device name** and **Property name**.

Buzzwords

- Technology Agnostic Protocols
- Late Binding
- Explicit Encapsulation
- Loose Coupling
- DevOps
- Microservices
- Containers
- Asynchronous Communication
- Log Management tools
- API gateway and discovery service

Architecture of LoGuide and LoFocus Services. LoScript





Service (MetaData Aggregator)

Service (LoFits)

Service(LoCam)INDI (Deinterlace)

Instrument ARC-CCD

Service(LoCam)INDI (Slit Viewing Cam)

Instrument Slit View Cam

ActiveMQ Broker

Service (RunScript) Focus, Guiding, Science

Service (FitStar)

Storage
Guide Script
Focus Script

Service (Logging)

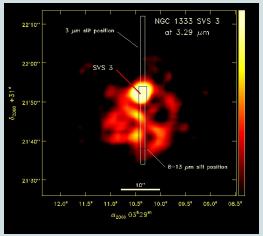
Services (LabView)

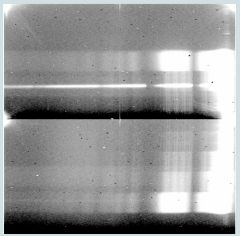
lucidchart.com

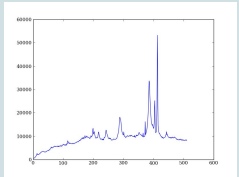
Google Scholar | Documents | Blank Page after Login to /documents – Lucidchart

Main Menu

Controls







Status