

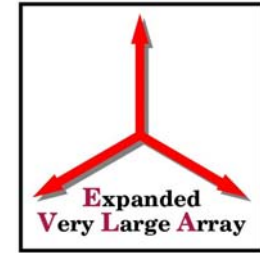
MCAF

(Metadata Capture and Formatting)

Rich Moeser



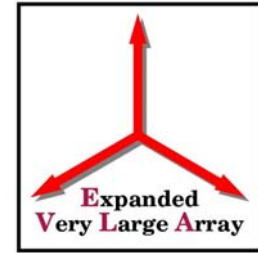
Outline



-
- MCAF overview
 - System diagram
 - Timeline
 - Status
 - Design
 - Possible Reuse (from IDCAF or ALMA)
 - SDM
 - Deployment
 - Summary/Questions



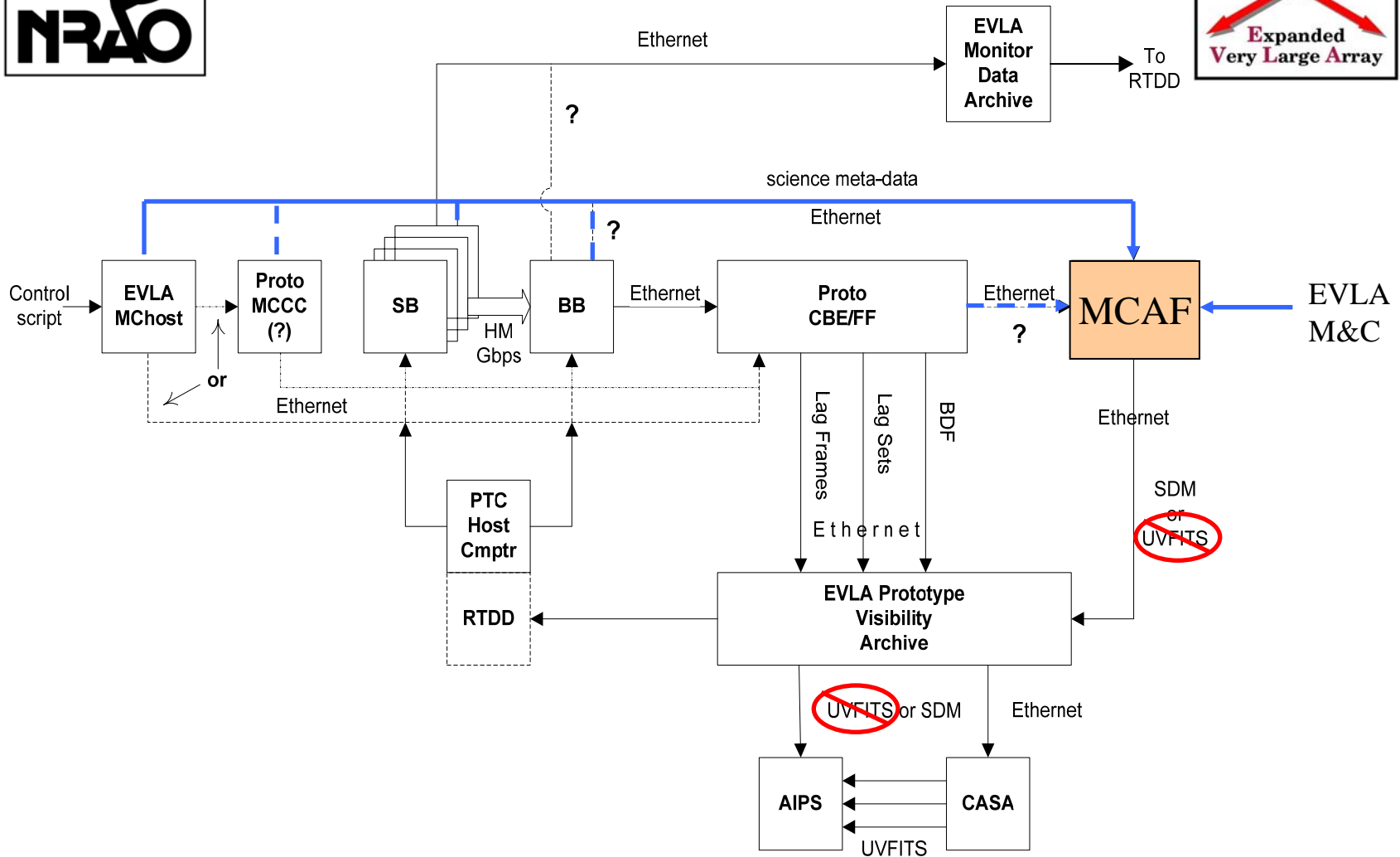
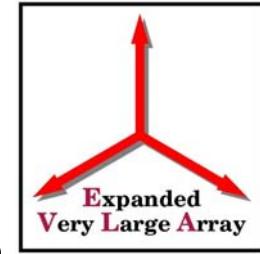
MCAF



- MCAF was originally named DCAF
 - The name was changed to emphasize that it's "meta" data that's being captured (as opposed to visibility or monitor data).
- MCAF's primary responsibilities are:
 - To collect science metadata from the EVLA system and the correlator.
 - Combine and reorganize the data
 - And write the data in ESDM (EVLA Science Data Model) format
- It is the successor of IDCAF (Interim Data Capture and Formatting)
 - This is the data capture process currently used by the EVLA system.
 - Required for the retirement of the Modcomp computers.
 - Writes the data in VLA export format to the VLA archive.
- Differences between IDCAF and MCAF
 - IDCAF = EMCS + VLA Correlator + VLA Export Format
 - MCAF = EMCS + WIDAR PTC + ESDM Format

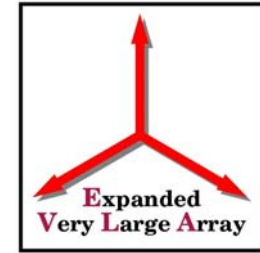


System Diagram





Phases/Timeline



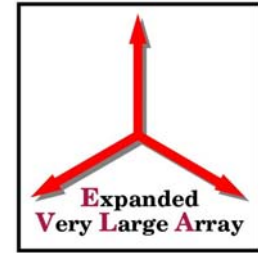
- Feb 2008
 - Detailed design complete
- Mar 2008
 - Schema definitions for all data going into MCAF
- May (early) 2008
 - Writing the minimal SDM (= mandatory tables)
- July (mid) 2008
 - Ready to support PTC tests
 - Writing minimal SDM + other required tables (if any)
- Q1 2009
 - Support for simple observing with WIDAR
 - “commissioning basic observing modes”
- Q4 2009
 - Writing full-blown SDM, supporting all tables for EVLA/WIDAR
 - “commissioning advanced observing modes”

ALMA deliverables: Q1 2008

- Typed XML schema
- Modifications to code generator to eliminate ACS/CORBA dependencies



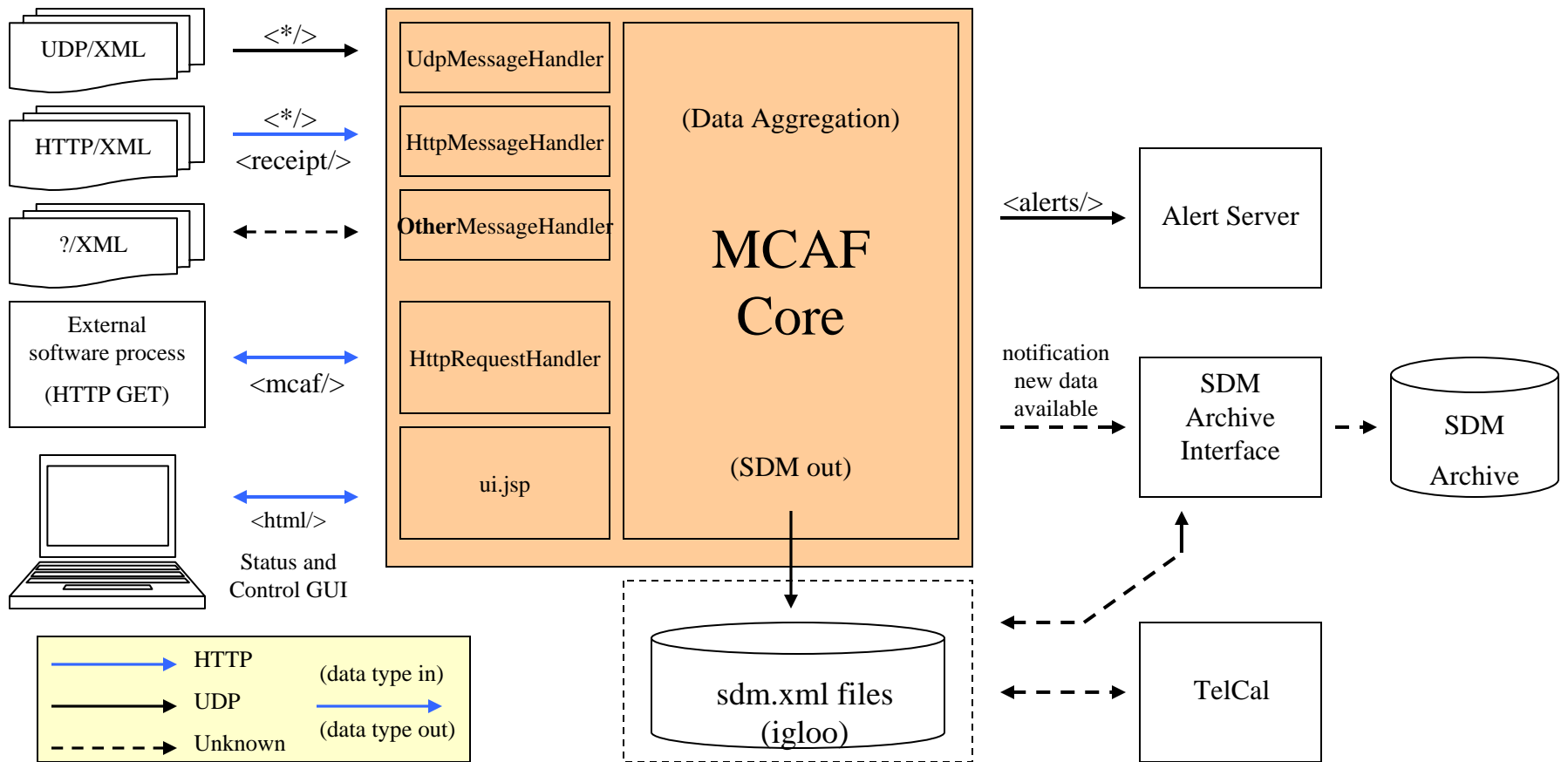
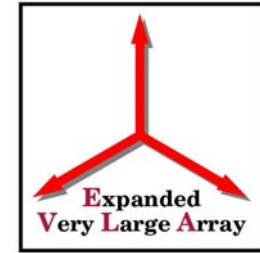
Status



-
- Currently in the early stages of design and prototype.
 - Focus has been mostly on infrastructure
 - Data collection
 - SDM handling (SDM to Java binding)
 - Communications

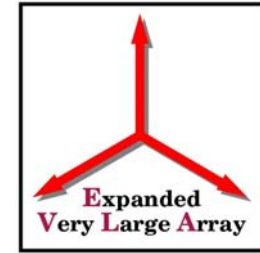


MCAF design





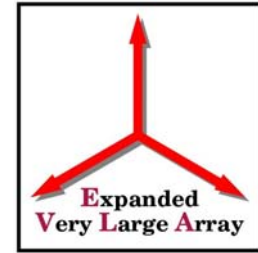
MCAF Input



-
- This includes all data required to build the ESDM
 - Data that is either sent to MCAF or data that it retrieves on its own.
-
- **EVLA Data Providers**
 - Executor
 - CMP
 - MIBs
 - Alert Server or Flagger
 - (depends on internal or external flagging)
 - Dynamic Scheduler (?)
 - TelCal(?)
 - Others?
 - **Correlator Data Providers**
 - CBE/FF (?)
 - Station Boards (?)
 - Baseline Boards (?)
 - MCCC (VCI) (?)
 - Others?



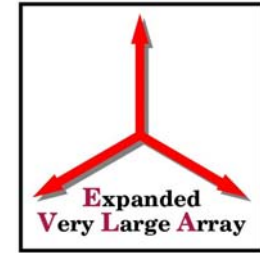
MCAF Output



-
- Writes SDM output files (to a staging disk)
 - Sends alerts to the Alert Server
 - Data to TelCal (the SDM files)
 - Archive notification (?)



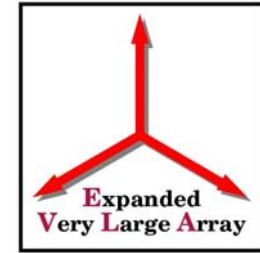
Communications



- Sending data to MCAF
 - UDP Multicasts containing XML documents
 - HTTP POSTs containing XML documents
 - Unknown data types received by MCAF will be logged and dropped.
- MCAF fetching data
 - What data will MCAF need to collect on its own?
 - If MCAF needs to get data from the SBs will it need to open a connection to each one?
 - What is the communications protocol?
- Access to MCAF information from clients (Java, python, ...)
 - HTTP interface to retrieve basic status information, e.g. health, number of packets received and processed, errors, start date and time, etc
 - Data will be an XML <mcaf> document back to the client.
- Browser access...
 - A simple JSP (Java Server Pages) will display MCAF status and possibly present control options (shutdown, restart, etc).



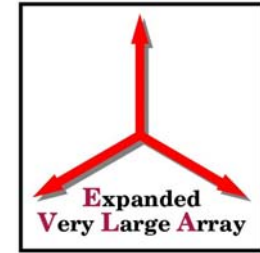
Possible Reuse? (from IDCAF or ALMA)



- IDCAF is written in C, MCAF will be written in Java.
 - No chance of reuse there
 - Possibility for reusing parts of the design
- The infrastructure for sending data to IDCAF exists, e.g. in the Executor, so there's a good chance the existing XML schema can be used and simply extended.
- There's a very good chance of being able to use ALMA's code for reading and writing SDM.



SDM

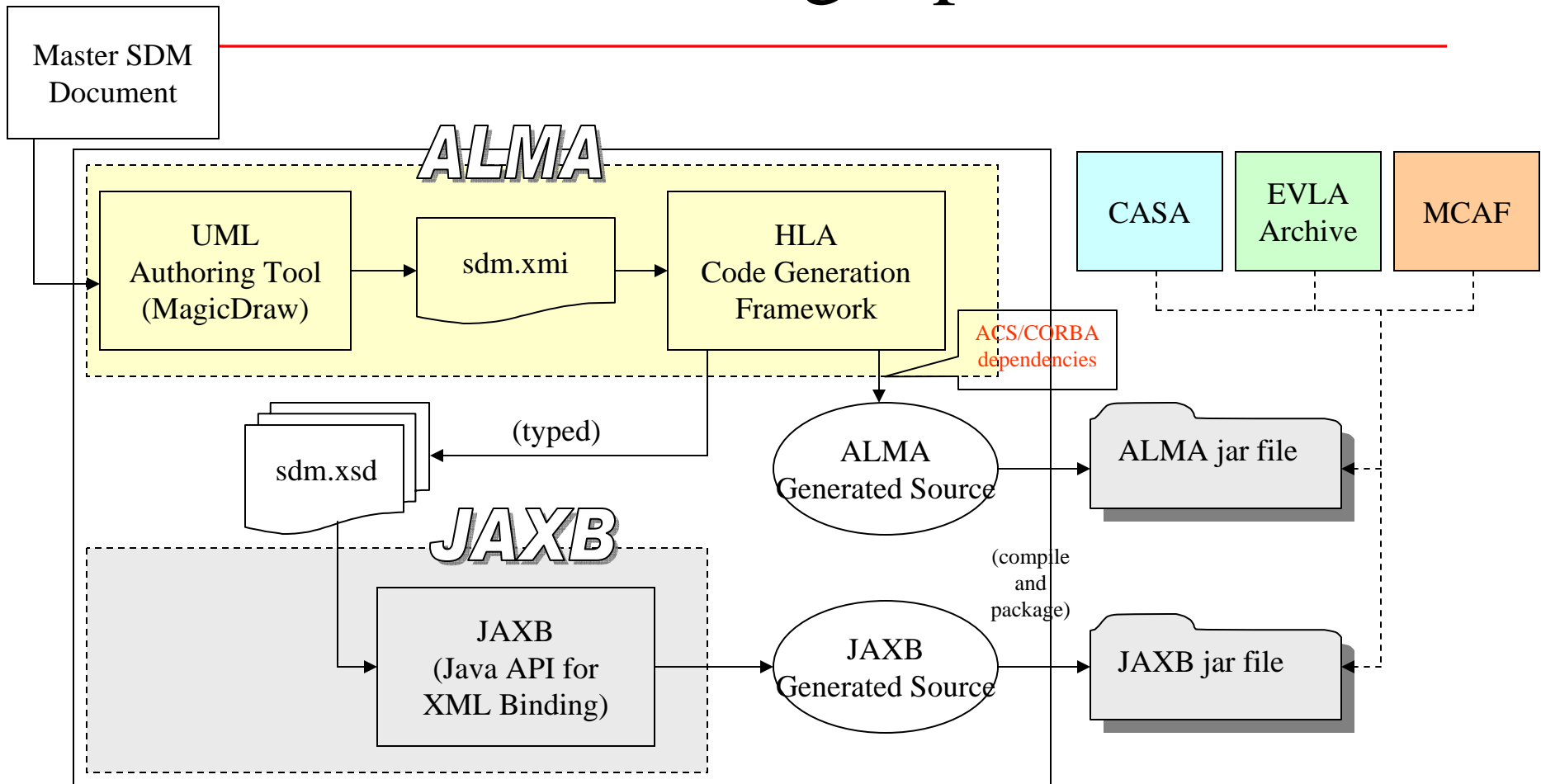
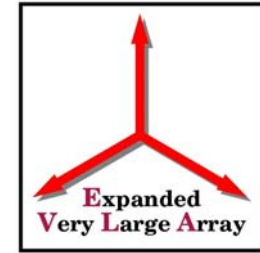


-
- The SDM is a collection of ~40 “tables” containing “rows” of data of a known type.
 - The “minimal” SDM will create and write the “mandatory” tables of the SDM.
 - The “mandatory” tables:
 - Main, AlmaCorrelatorMode, Antenna, ConfigDescription, DataDescription, Feed, Field, Polarization, Processor, Scan, Source, SpectralWindow, State, SubScan
 - The remainder of the table are considered “optional” tables:
 - Beam, CalDevice, CalAtmosphere, Doppler, Ephemeris, ExecBlock, Focus, FocusModel, GainTracking, History, Observation, Pointing, PointingModel, Receiver, SBSummary, Seeing, SourceParameter, SquareLawDetector, Station, SwitchCycle, TotalPowerData, WVMcal, Weather.
 - Are the mandatory tables sufficient for the PTC tests? If not, what else is needed?



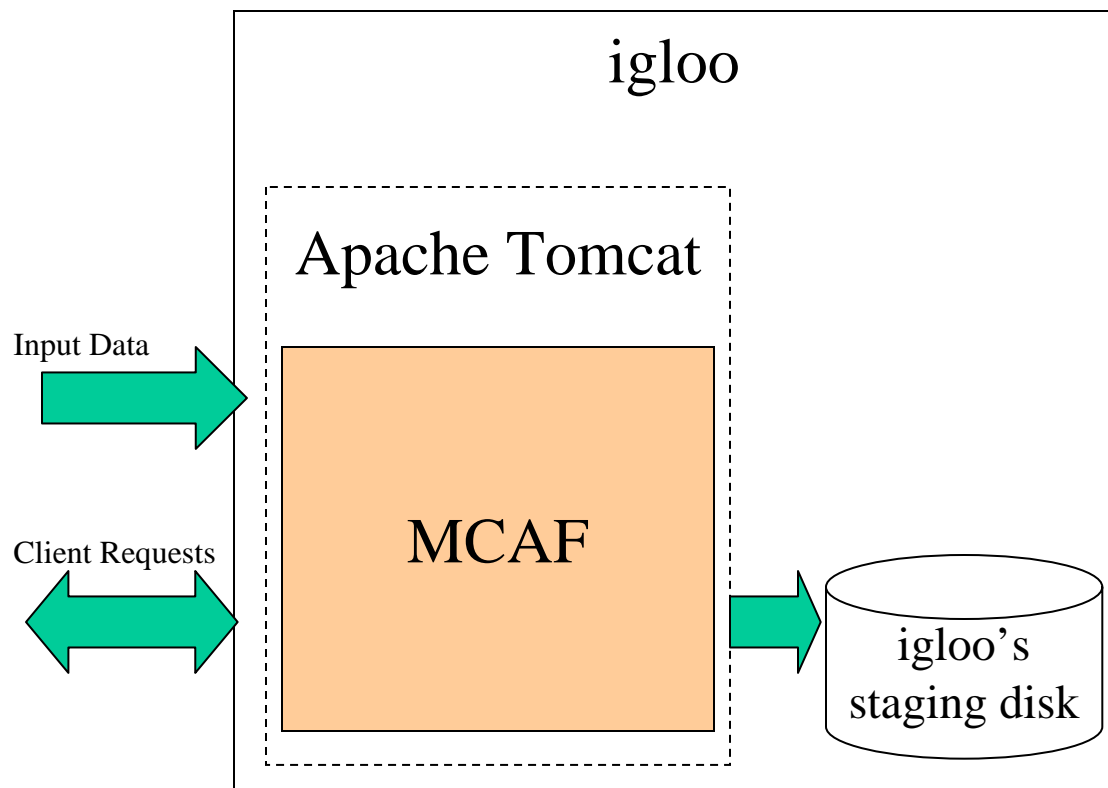
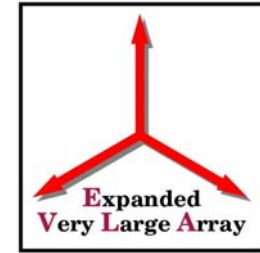
SDM

Java Binding Options





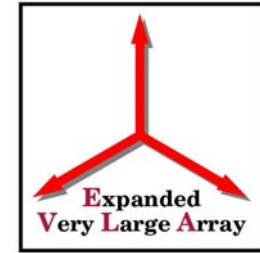
MCAF Deployment for the PTC



- MCAF will run on the machine *igloo* (or its replacement) out at the site.
- It will be packaged as a “.war” file and deployed to a running instance of Apache Tomcat.
- If the system goes down for whatever reason Tomcat will automatically start and launch all of the applications in its container, including MCAF.



Summary/Questions



Summary

- Still in the early stages of MCAF
- A detailed examination is needed to find out where all of the data required by the SDM will originate
- A first cut that writes minimal SDM will be ready by May
- There's a pretty good chance we'll be able to use some of ALMA's software.
- Determine whether or not flagging should be built into MCAF or if it should be a standalone external process.

Questions

- Does MCAF gather anything directly from the CBE/FF/SBs/BBs/MCCC?
 - What is the communications protocol?
- Is the “minimal” SDM sufficient for PTC testing?
- The plan is to have MCAF writing SDM in May. Is that acceptable?