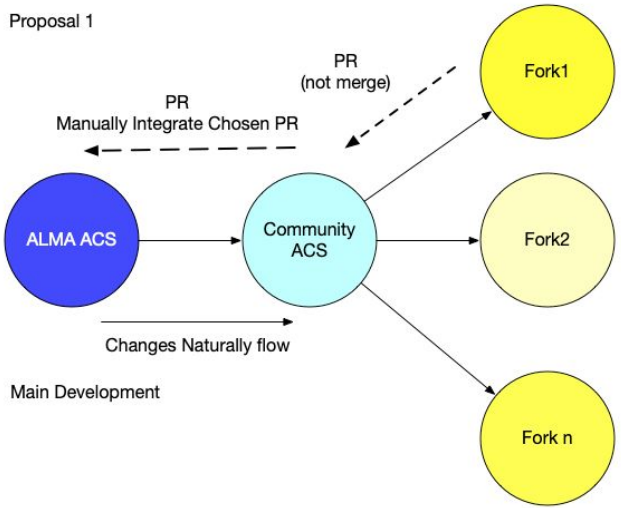
ACS Community

Who are we?

- ALMA
- APEX
- CTA / ASTRI
- LLAMA
- RT40m-ARIES
- HESS
- Anyone one else interested, really

How does the community work?

- ALMA is leading ACS Maintenance (2 FTEs) and Development (Best Effort)
  - Focused on ALMA's priorities
- Preparing releases and making them available to the community



How does the community work?

- Receiving questions, requests and suggestions from community
- Receiving patches and integrating them in ACS
- Creating tickets, following up and resolving them
- Organization of community meetings

What topics were discussed regarding ACS’ Future?

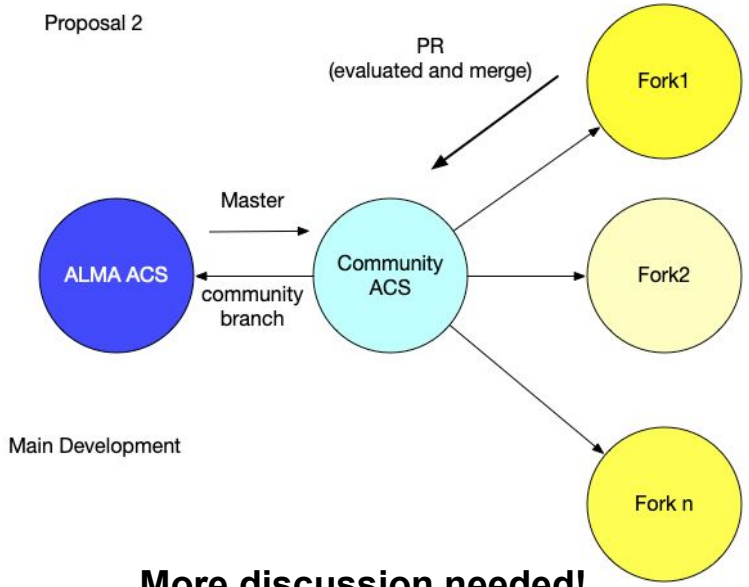
- Issue Tracking and Integration
- ACS Whish List
- Official Docker Image
- BulkData Technology for the Community version
- ACSThreads Improvements
- Notification Service (Notification Channel) Replacement
- Component Caller / Event Converter Technology Abstraction

Issue Tracking and Integration

Issue Tracking Alternatives:

- JIRA Open Source License
  - Create public project (read-only)
  - Community Members can write
  - Integrate users from the community
- ALMA and Community Integration
  - ALMA ACS ‘acs/community’ branch
  - Community ACS ‘master’
  - Release branches
- Alternatives
  - Main development on Community ACS
  - No ALMA ACS Repository

Integration:



More discussion needed!

Issue Tracking and Integration

- JIRA is OK if users management is done efficiently
  - Some problems in the past
- Inclination towards ACS Community as main repository
  - Steering committee
  - Possible political effects on ALMA
- See how other projects are managing shared frameworks/software (Tango?)
- Should ACS Community repository have write permissions?
  - Sure, but for whom? What branches? Etc.
  - It would require a development/delivery process

ACS Wish List

- Native support of OPC UA
- Support of TAI for timestamps
- Integration with a database for more than 32 items history for an attribute
- Fully Java BACI support
- Fully Python BACI support
- Structure BACI properties (even if basic / with limitations)
- Simpler CDB:
  - The CDB XML interface feels very clumsy and not well documented.
  - Is there a way to give beginners a simpler file-based CDB for testing which is maybe based on something like YAML, TOML, JSON, INI or so? I guess the XML was chosen, so that the schemata can be validated, but is that something beginners even need?

ACS Wish List

- Add security features:
  - Authentication and authorization
  - Session binding
  - Encryption
- Modification of omniorb to include python command-line help documentation
- getTemplate improvement: The interface of the "getTemplate" CLI tool is very strange. No tab completion, no help
- Features that may exist but not known or documented, or not exist yet
  - How can we get the contents of the "acscommandcenter" in a Python? In particular, how can we start containers, stop containers, check which containers are running, from Python?

ACS Wish List

- ACS shipping
  - Slim-down ACS dependencies - e.g. Python, Python Libraries, ...ACS & Python
  - Can we get ACS to use the System Python instead of shipping their own
  - Use stdlib Python logging, while ACS just hooks an "ACSHandler" into the stdlib python
  - Enumerations in IDL, they do not get properly translated into Python Enums (If possible, likely not)
  - Improving the installation process via graphical UI. The end would be working ACS incl. an introot etc.
  - The docker and vagrant are steps toward that
  - Not hight priority, but would invite more participants.

ACS Wish List

- Performance enhancement
  - "acsStart" and "acsStop" optimization. They take forever, this should not need so long
  - Why does it take so much memory?
    - 1st start of ACS needs to have at least 3-4 GB (TB)
  - "getDynamicComponent" is a blocking call, that waits for the remote component to be actually loaded. This is not necessary, it could just create the "objref" and return right away independent on how long the component needs to start. Can we fix that?
  - Better handling of attributes subscription, particularly when client disconnect
  - The creation of mockup components within the same process could be optimized - needs investigation
  - Be able to create a test with mock-up components without the need to define IDLs
  - Could try to use the Python Component simulator (TBC). Can be used without ILD or test
  - Action item: have a look

ACS Wish List

- ACS technologies you'd think needs a replacement (please indicate the reason: obsolescence, security, ...)
  - Old-style makefile
  - ACS build system
    - Separate ExtProds from ACS core
    - Could come from the operating system or EPEL - someone needs to maintain this
  - Use of modern unit-test frameworks instead of TAT
  - DDS (see yesterday discussion)
  - How long can we maintain Java SWING UIs?
    - ACS team Thinking to replace to pyQT
      - Included nice tool for graphing pyQTGraph
      - Issues - another dependency - will limit the libraries dependency
    - Alternative: web interface
      - ALMA and CTA looking at UI web-based technologies
  - Not urgent, but alternatives to CORBA should be studied

ACS Wish List

- Enhancing the ACS experience
  - Better documentation
  - Link with ticketing - will come very soon (couple of weeks)
  - Integrating with tutorials from this workshop
  - Means of communication - Slack, Ticketing, …
  - More frequent community calls
  - Better and more sophisticated examples

Official Docker Image

- Ready-made minimal Docker Image
- Quick and easy way for newcomers!
- Explore ACS Examples
- Develop and test first component
- Extendable with user experience

Current Prototype

- https://hub.docker.com/r/dneise/acs_test
- https://github.com/dneise/acs_test

BulkData Technology for the Community

- ALMA Uses Proprietary (RTI) DDS
  - Not available for community
- ACS BulkData implementation on CORBA A/V Streaming
  - Maintenance was dropped years ago
  - It had reliability and robustness issues
- Prototype implemented by UTFSM for CTA
  - ZeroMQ-based Implementation
  - Compatible with ACS' BulkData API
  - Open Source
  - Still a prototype

ACS Threads Improvements

- Initial motivations to implement ACS Threads in C++
  - Portability, abstraction for complex interfaces, configuration capabilities
- While them may no longer be concerns:
  - Portability: std::thread, boost::thread
  - Abstraction for complex interfaces: std::thread, boost::thread
  - Configuration capabilities: boost::thread
- Setting up the logger for each created thread
- Thread Manager resources deallocation on component / container exit
  - One alternative would be to design a registration for logging and lifecycle management
- This could also be desirable for other language implementations

Notification Channel Replacement

- Reliability Problem
- Hard to debug and retrieve internal information
- Obsolescence management plan
- Plans to study alternatives
  - Kafka, ActiveMQ, Akka Streams, ZeroMQ, etc.
- Plan to implement and replace CORBA CosNotification service

Component Caller / Event Converter Abstraction

- Component Caller: Proxy to access components from REST API
  - Generic interface receiving JSON data with component, method and arguments information
  - Generic response in JSON format, including complex IDL data structures
- Event Converter: Converter from Notification Channel events to Redis queues
  - Events IDL structs are converted to JSON format and pushed into Redis
  - Support for Reliable and Unreliable Subscribers
- Allows to move ACS interactions to Web UI
- Current implementation is fixed into REST API and Redis
  - Bridge design pattern or a similar one could be used to abstract the underlying technology
- Pending work
  - Callbacks handling
  - Security

ACS Community Objectives

- Increase Community Collaboration
  - Identify Current Community
  - Public Repository
  - Issue Tracking
  - Releases
  - Integration
  - Building / Packaging / Distributing
- Increase Community Engagement
  - More frequent community meetings
  - Better means of communication (Slack, Issue Tracking, etc.)

ACS Community Objectives

- Improve ACS Visibility
  - Website + Confluence
  - Generated Documentation (Doxygen + Javadoc + etc.)
  - Handmade Documentation (Technical Documents + Tutorials + Hand Books + etc.)
  - Updated Frequently Asked Questions
  - DockerHub Official Docker Image + Dev Images
  - ACS Community Slack Page
- Modernize the Framework
  - Replacement of technologies
  - New developments
  - Support more and new technologies
  - Improve performance

Next Steps...

- Workshop Follow Up Meeting
  - Close action items
  - Define collaboration strategies for specific projects or developments
  - Discuss topics that didn’t fit into workshop’s timeframe
  - Try to come up with action plans for more complex decisions
- Define Community’s Strategic View
  - Strategies
  - Mission
  - Vision