ACS Community Roadmap

Jorge Avarias* et al.

November 4, 2014

1 Introduction

This ACS community development roadmap has been prepared as a recommendation by the authors of this document, to guide an initial discussion. It is based on previous community initiatives, personal experience and general user feedback, and does not impose any specific actions on ACS community participants.

Defining a final version of this roadmap, based on requirements and resources offered by the community should be a short-term task for the ACB [1].

2 Proposed Development Areas

2.1 Platform/OS Support

- Porting to other platforms, besides RHEL/CentOS 6 and 7. This may consider 32 and 64-bit support for the other platform.
- The ACB (see community organization: `https://github.com/ACS-Community/ACS/wiki/ACS-Community-Organization`) must decide what platforms should be supported.
- For now the suggestion is to port ACS to other mainstream linux platforms. e.g. Fedora, Ubuntu, Debian, OpenSUSE?

2.2 ACS Packaging

- Repackaging of ACS into modular pieces is desirable, discarding the external products with packages provided by the Linux distribution, if they are not available then create the respective packages.

*Not as NRAO employee

- New building system, needed to improve the automatic dependency resolution among different modules.
- At the end ACS should be packaged and deployed as any other Linux binary, based on each distribution guidelines.

2.3 Embedded Systems

- Porting to embedded systems. We understand as embedded systems small boards without enough RAM or processor power to run all ACS Services. Mainstream embedded platforms should be identified as targets.
- Study and develop a common procedure to port ACS to such systems, regardless of the cross-compiling technique.
- Profile the container and client side of ACS, to minimize memory footprint and to be efficient with the CPU usage for such systems.
- Prepare a DevIO code base (like community provided drivers), so users can speed up the development with hardware already tested by the community.

2.4 ACS Services

- Define metrics to measure the reliability of services.
- Profile and audit the code of essential services.
- Explore alternatives according to each service and its current state, if necessary.
- Monitoring tools for services.

2.5 Development Tools

- ACS currently lacks proper development tool integration. An IDE could be identified to help to flat the learning curve for the newcomer ACS developers. The IDE must support all ACS-supported languages and component/client code skeleton templates.
- As part of the IDE, provide an editor to simplify the deployment of ACS in testing environments. Perhaps an integration of the acscommandcenter, cdbEditors, alarmEditor, etc. to an IDE could be sufficient.
- One-click deployment into a target embedded system.

- Create an automated ACS testing environment, i.e. identify suitable TAT replacement.

2.6 Documentation

- Better documentation for the packages provided by ACS. Clarify the "Control" and programming model that ACS provides.
- Stand-alone crash course for beginners.
- Building guide.
- Create coding standards for each supported language. ALMA coding standards were used in the initial development, and could be adapted for future guidelines.

2.7 ACS Community Services

- Build service. Not just for ACS, but also for open-source ACS related projects.
- Official binary distribution repository and mirrors.
- Linux distribution repositories for "thrid-party" packages no included in the official distribution, and for ACS modules/packages.

3 Roadmap

At this point it is not clear what specific features or functionality are required by each ACS Community participating project, nor what the plans of all contributing developers are. Therefore the roadmap presented here only defines milestones not bound to dates, and could be subject to modifications, especially in later milestones.

Milestone 1 Re-organize ACS distribution and provide a basic build and testing system

- CentOS 6 has been selected by the ACB as the main development platform given the current building and testing infrastructure. A build and test server with this OS version is currently running in a machine lend by UTFSM University in Chile. However, the community might provide support for other Linux distributions in the form of patches and infrastructure to support building/testing.

- External products might be used from the distro repositories. Developers should not need to maintain custom packages unless it is strictly necessary. For external products that are not part of the distribution, we shall provide an RPM (or other package) to install as a dependency.
- Configure the build server to also deliver test results. These configurations should be a deliverable of every new porting effort.
- Basic installation and configuration documentation for newcomer developers.

Milestone 2 Move ACS to newer and different platforms

- Create specific community use case tutorials.
- ACS has a history of being officially built for RHEL. Those are stable platforms used for enterprise deployments. However, not all community participants want to use those, and there are other suitable options on the market (e.g. Ubuntu LTS, Debian Stable, OpenSUSE Evergreen, etc). Support should be provided for the platforms most used within the ACS community participants. At the same time, old versions should no longer be maintained, aiming to support the newest (or newer) version/distro.
- Move to CentOS 7 and support at least one other linux distribution (incl. building and testing).
- Provide clear instructions of how to port ACS to embedded system running Linux. It is suggested to focus the development on ARM platforms, as they are quite widespread and have a lot of community support.
- Profiling of the client side of ACS to identify weaknesses.

Milestone 3 Advancing ACS

- Study ACS services' reliability.
- New ACS build tools.
- First version of ACS Integrate Development Environment (IDE).

Milestone 4 TBD

References

[1] ACS Community, *ACS Community Organization*, nov, 2014. `https://github.com/ACS-Community/ACS/wiki/ACS-Community-Organization`