# FuSa SIG September, 3rd 2019

Agenda items are added black.
Minutes are added in blue.
Closed ACTIONS in green.
Open ACTIONS in red.

## A) Link to recorded sessions

https://citrix.sharefile.com/d-s9ca123d544e46818

## B) Attendees (right now based on past attendees - delete/add as appropriate)

- **Antonio Priore**
- Julien Grall
- Robin Randhawa
- George Dunlap
- **Lars Kurth**
- **Alex Agizim**, Artem Mygaiev
- **Kate Stewart**
- **Shinya Konishi,** Hisao Munakata
- **Francesco Brancati**
- Stefano Stabellini
- Piotr Serwa
- Robert Heinen
- **David Ward**
- Claudio Gregorio
- Christopher Zimmer
- Vasco Fachin

*Above is a list of past regular attendees. UPDATE*

## C) Actions not yet resolved

**High:** Lars to set up a smaller meeting with Francesco Brancati, Kate, Artem & Vasco to walk through the test infrastructure we have. Will talk to Ian Jackson and/or other stake-holders.

Initial chat with Ian and Andrew. Will follow up with more detail.
Agreed.
Ian requested a set of questions, to make the session more productive **before** the meeting.
There have also been changes to how we approach CI, which will create new opportunities
Open questions on how much testing needs to be done on real Hardware vs. e.g. in a simulated

environment via QEMU (e.g. Zephyr is mostly doing simulation, while AGL uses new LAVA versions)

Franceso reviewed the material
Note: Lars cannot set up a call for the next two weeks
Kate in Europe next week, may be difficult

**Medium:** Julien to talk to Lava owner on state and usability

Julien: I spoke with one of the Lava owner last Friday. It looks like they have done some works to use QEMU and KVM in Lava 2 a couple of years ago. We could possibly re-use what has been done for Xen. I haven't had yet the chance to speak with Wookey who did the work.

**Medium:** Lars to put together a proposal on coding standards and checking tools as a test-case for MISRA

Lars: Have not had time to do this yet and wont be able to do so in the next 2 weeks

~~**Medium**: Francesco to send Kate links or edit https://docs.google.com/document/d/1OPRFHtyX8ASU831Db_YTOB_LYOBURC-3VPv6vjy8ijk/edit#~~

**Medium**: Francesco to collate information from relevant ISO documents and share with group
Was sent to Artem
Did not discuss, please send to list

## D) Proposed Agenda Items

### D.1) Brief updates from stream leaders

I don't expect much as many just have come back from holidays

**Artem:** continued trying to parse mailing list data using markmail.org (create a whole chain of e-mail threads) covering technical discussions before a change
=> That allows us to understand requirements from public discussions
Used markmail API / Database
For most changesets: what happened, who authorized, who reviewed, why and linkage to changeset - security items are not covered
Did do some code coverage stuff: aka for what code was reviewed
Looking for tool to visualize to allow us to more easily condense

Kate: CRAIGIT (https://cregit.linuxsources.org/) project may be useful
Artem: Looked at it, but does not seem to connect different iterations - done by another research group.
ACTION: Kate can put Artem in touch with them (connecting CRAIGIT with Lore and that work) - Artem to initiate

Kate: there is some visualization stuff in CRAIGIT

**Kate:** Please add tools to TOOLS List

**Francesco**
Asked the team to start using **understand** tool (scitools) on current version of Xen
Request for help from Stefano or Artem
Antonio: asks if understand is capable of MISRA C checking

### D.2) Compatibility of tooling environment with safety standards

Generally, the common theme to enable safety in Xen and open source projects in general is that the **workflow cannot be impacted.** Breaking this down the core-interaction a developer would have with safety aspects of development this would mean

> *There can't be additional tools a developer has to use in PARALLEL to the existing workflow: for example, having to go to a separate web portal or UI tool to check MISRA violations, manage requirements outside the code using a client/server based tool, manage project wide changes (aka change management), etc. won't work.*

**E.g. looking at MISRA:**

- Ideally, we want a tool that works like a compiler locally and is free
- But failing this, a tool that is integrated into the code submission and review process would work: for example, today when someone submits a change we don't expect that the developers test their changes against all compiler and Linux distro combinations. But when a change is submitted CI machinery kicks into gear doing this, essentially blocking the change if issues are found. MISRA validation tools would have to be able to run in such an environment and would also have to be applicable to changesets applied to the existing baseline

**E.g. looking at change management**

- If all generated artefacts were stored in the source tree, change management would just be handled in the normal way via git commits
- If other systems such as user stories, issue trackers, etc are also used, there would also need to be linkage between git commits and user stories, JIRA issues, etc.
- Artem also has experimented with making e-mail based code review more accessible by extracting information from markmail: the primary use-case is to help create documentation (such as requirements) from existing material, which today is not easily manageable. But also to help prove that we followed the code review process (aka almost like code coverage

for reviews) and address weak areas AND to improva linkage between code review and code

In other words, the extra layer of change management that is normally required to keep artefacts in sync that are stored in different systems, basically goes away

**E.g. looking at requirements, document and traceability management**

- Ideally artefacts would be stored in the source tree and in some cases in the code itself. The smaller the “distance” to the code, the better. This should be achievable with appropriate tooling
- Notionally the process of keeping a traceability matrix up-to-date, should be similar to keeping dependencies in the code updated. In other words, a tool which behaves very much like a “linker” would for example fit into this paradigm

  **Specifics for https://github.com/doorstop-dev/doorstop**

  - In principle the tool is suitable with some changes
  - There are some concerns about the long-term viability of the project: I was planning to raise this, as well as other tools, at the ELISA workshop
  - The basic idea would be to use the tool to manage dependencies between documentation and code artefacts. The tool as-is allows tracking a hierarchy of pieces of documentation and creating dependency (traceability) reports. In practice this means that if something high up the tree changes, the user has to confirm that everything which is dependent on a requirement has to be verified as still valid or modified, should something higher up change.

  - The downside of the tool as-is, is that it is fairly rigid when it comes to encoding the artefacts: they are stored in the source tree, but right now: the formatting is in a form which I don’t think can get past any open source developer community, it requires one requirement/document is stored in one source file, it does not allow embedding documentation in source code – in any case, this seems fixable with 2-3 weeks of effort

**Testing seems straightforward** with what was proposed before: the only potential wrinkle is in that I cannot see a model where ALL testing will be done by the project

- I am looking at a model where we end up with infrastructure that allows the project to run some tests but allows a vendor (or multiple ones) to integrate with our CI and comment on changes or even block changes that lead to test failures

Need validation that the proposed approach can be made to fit safety standards (details can be sorted out in due course).

It would also be important to know what level of proof and maybe metrics we would need to collect to prove that we satisfy the development process requirements of safety standards.

Antonio: what is the confidence of not missing anything?
Antonio: need to demonstrate that what we say, is fulfilled. Need to show that checks still hold, if done in the past

Process confidence: need to prove that all the steps have been done

David: Need to define what we need by compliance (in terms of what we need / what we dont need to follow)
David: We will need to know which rules we have / may not follow and record deviations

Lars: We had covered this in the past, but are currently blocked on the certificatioun route which impacts

ACTION: Lars to follow up on Bugseng

### D.3) Planning/preparing for ELISA workshop

Lars: submitted tooling discussion. Not yet quite sure how to approach this
Ideally, I would end up with some agreement that attendees come up with a plan to

a) Recommend some tools
b) Find funding ($$s or manpower) to keep alive / evolve projects such as DOORSTOP, which are critical to safety certication for open source

Discussed whether this is a good step for the ELISA workshop and there was agreement

## E) AOB

1. Idea: Lars to write ELISA Workshop event report and publish it as contributed article somewhere - CHECK WITH KATE WHETHER OK [compatibility with Chatham House rules?]