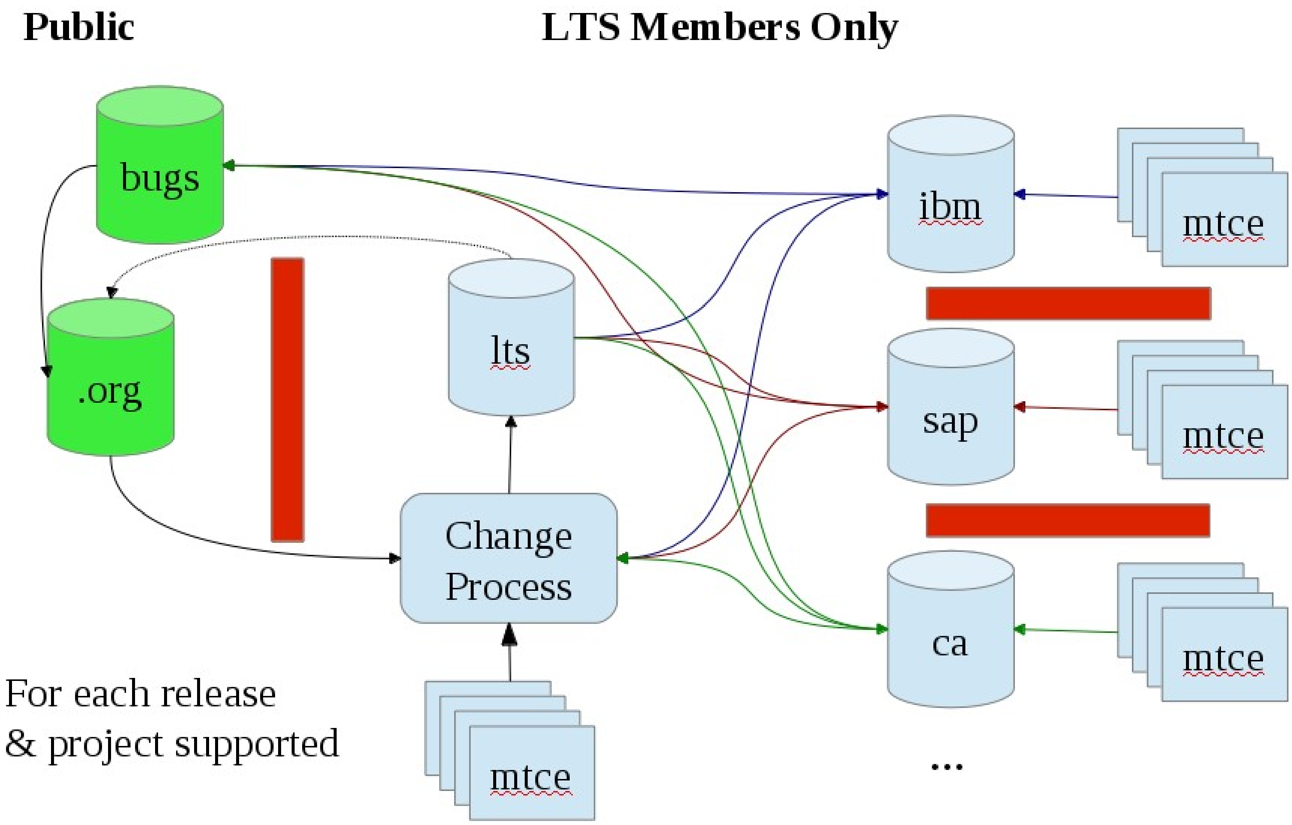
LTS use of Gerrit

This brief document is to describe how Gerrit is used in LTS.

Gerrit is a powerful open source contribution management system and change review system built on top of git. For those familiar with Github's pull request review/commenting features or Reviewboard, Gerrit is very similar. The most obvious feature you'll see is side-by-side code review with colour coding and even some basic syntax highlighting. People can comment and vote on the code changes contributed.

LTS provides each Steering Committee or Premium member with a private company specific forge. LTS members collaborate in the LTS central forge which the members share. The following diagram provides an overview of the LTS forge.



Maintenance committers are appointed by SC or Premium LTS members to complete code changes/fixes on their behalf.

Gerrit provides the authorization and authentication for all of LTS. The account database for maintenance committers is persisted in LDAP and allows maintenance committers to re-use their Eclipse ids and passwords.

Company specific forges are actually private branches within a single git repository. Branches for each company specific forge will be prefixed with the company name and only visible/accessible to that company's maintenance committers. Tags made in a company specific forge will only be visible to maintenance committers from that company.

Company Specific Forges

The rules for company specific forges are intentionally kept as simple as possible. Any maintenance committer can push direct to their respective company specific forge or optionally post a change for review with their peers from that company. These reviews are private to that company's maintenance committers.

There is a significant cost associated with working in the company specific forges. LTS member companies are responsible for keeping their forge (company specific branches) up to date. This requires inward rebasing or merging of content from LTS Central or the community as needed. Any merge conflicts must be solved by the company's maintenance committer(s). Also, all changes distributed from a company specific forge must be made available to comply with the EPL. This is a manual process to post the change for review in LTS Central's Gerrit instance and/or into Bugzilla.

LTS Central

LTS central will be set up as a shared forge called "ltscentral".

LTS Central will be visible as the root of LTS's Gerrit instance. Maintenance committers can navigate to access a listing of company specific branches for their company if needed.

All maintenance committers will have access to all of LTS Central. We do not differentiate maintenance committers on a per project basis.

All tags and branches for LTS Central will be visible to all maintenance committers.

For LTS central, assuming the company specific forges satisfy the urgent/ emergency use case(s), rules for LTS Central are more stringent as follows.

For LTS central

- Any Eclipse community committer for any project (hosted at eclipse.org) can vote +1/-1 and comment on any change being reviewed for LTS central.

These votes & comments are informational and do not trigger submit nor block submission in LTS Central.

AND:

Review is mandatory in LTS Central

- A change pushed for review in LTS Central triggers a Hudson build & tests. If these are successful, the change will be labeled verified. A change cannot be submitted until it is verified.
- Any maintenance committer for any company can vote +2/-2 and comment on any change being reviewed for LTS central.
- Any -2 votes by any maintenance committers blocks being able to submit a change.
- M number of +2 votes and no -2 votes enables submit. M will be 2 to start but may be changed by the SC as they see fit.