

SAT Meeting 2014-09-23

Date

September 23, 2014 12:00-2:00 PDT




< [previous \(2014-09-16\)](#) | [next \(...\)](#) >

Attendees

Kian-Tat Lim	<input checked="" type="checkbox"/>
Mario Juric	<input checked="" type="checkbox"/>
Jacek Becla	<input checked="" type="checkbox"/>
Unknown User (ciardi)	<input type="checkbox"/>
Gregory Dubois-Felsmann	<input checked="" type="checkbox"/>
Donald Petravick	<input type="checkbox"/> (not available)
Frossie Economou	<input checked="" type="checkbox"/>
Unknown User (rlambert)	<input type="checkbox"/>
Robert Lupton	<input checked="" type="checkbox"/>
Unknown User (xiuqin)	<input checked="" type="checkbox"/>
Simon Krughoff	<input checked="" type="checkbox"/> (now official UW rep)

Discussion Items

Item	Who	Notes
------	-----	-------

Discussion	KTL	<p>Winter2015 Package Reorganization Planning</p> <ul style="list-style-type: none"> BaBar had many packages (3.5MLOC, 40% in core library, rest application-specific) <ul style="list-style-type: none"> Had 1000 packages in production builds; easy to create new ones; made it easy to manage dependencies 5 sub-pipelines doing completely different things, sharing only limited common code (10%) Evolved that way; could possibly have worked with a different approach Did not work with one package per area of responsibility; needed to use them for dependency control Could work within system touching only a few of the packages Working within shared environment; global build provided most of code; developers only changed a little Testing depended on stack of libraries, but those were already built Also had tool to test effects of lower changes on upper packages Renamings weren't particularly difficult Abstraction/interface + implementation(s) led to separate packages No one person knew whole picture, though Hard to take a slice of the code and deliver it to people <ul style="list-style-type: none"> Could have used a coarser-grained organization on top of fine-grained Frossie: want limited number to improve adoptability by outsiders <ul style="list-style-type: none"> Agree with interface/implementation split Want to be able to test packages in isolation; not install lots of dependencies to build/test a single one Test control flow of tasks without science by stubbing/mocking it out <ul style="list-style-type: none"> GPDP: But can't stub out all science and still be useful But afw in particular is a library with a wide interface that is hard to mock out Linux kernel model: <ul style="list-style-type: none"> Build via a hierarchy of drivers Stack should be thought of as a Linux/Anaconda distribution: install whole thing or parts Or are we more like PyPI/CPAN/Ruby gems? Repos and packages don't have to be aligned GPDP: Class-rich data objects lead to complex dependencies <ul style="list-style-type: none"> As opposed to something like byte streams GPDP: Algorithms built as layers of functionality also lead to more dependencies Issue is more distribution of packages and making them easily used for development (and not forcing rebuilding of the entire stack)
Action Request	KTL	<div>  DM-934 - Jira project doesn't exist or you don't have permission to view it. </div> <p>; with further update, can this be recommended?</p> <ul style="list-style-type: none"> No discussion or action
Action Request	Frossie	<div>  DM-1203 - Jira project doesn't exist or you don't have permission to view it. </div> <ul style="list-style-type: none"> Want fewer timezones overall, particularly between system logs and application logs Unclear whether better to shift the application logs to GMT or system logs to local time Tentative conclusion: probably best to try to get all logs into GMT Also crontabs and other things? Still a potential issue with TAI and UTC, which are a lot easier to confuse than different timezones
Discussion	KTL	<p>How legalistic should the SAT be?</p> <ul style="list-style-type: none"> No discussion or action
Design Deep Dive	KTL	<div>  DM-35 - Jira project doesn't exist or you don't have permission to view it. </div> <p>Alert Generation and Distribution</p> <ul style="list-style-type: none"> People were not prepared to follow up Mario promises to look over this by Thursday

Design Deep Dive	KTL	Data Release Production Overview <ul style="list-style-type: none">• No discussion• Mario promises to look over this by Thursday
------------------	-----	---

Action Items

- ☒ Mario Juric and Robert Lupton: comment on Alert Generation and Distribution and Data Release Production Overview by the end of the week.
- ☐ All: Think further about package reorganization.
- ☒ Frossie Economou: Think about an RPM model for distributing, using, and developing the Stack.