

cwebber	— MEETING BEGINS —	
	Ok, so on the agenda...	9:09 am
holoway_	hi	9:09 am
cwebber	holoway_: just in time	9:09 am
	Agenda can be found at https://github.com/opscode/chef-community-irc-m...	9:09 am
	<i>holoway_ is now known as holoway</i>	9:10 am
holoway	we have a bunch of stuff today - sorry I'm late	9:11 am
	token auth, root aliases, powershell, audit mode, osx shennanigans and the return of dialects	9:11 am
	let us start with otkens	9:12 am
	https://github.com/opscode/chef-rfc/pull/65	9:12 am
coderanger	So the tl;dr is adding a new auth mode alongside keys+signatures	9:12 am
holoway	coderanger: you want to give us the tl;dr (jinx)	9:12 am
coderanger	To allow for cases where keys are impractical like delegated auth or web clients	9:13 am
holoway	I think we pretty much all believe this is a universal win, and we have some caveats about specifics of the API (in particular, where policy on expiration can be set, etc.)	9:13 am
kallistec	my only question is if some of what's specified overlaps with something like oauth2 that we could delegate to.	9:14 am
coderanger	The two uses cases from the summit that spawned this are making the Web UI proxy layer much thinner, and building objcap security systems that work with Chef	9:14 am

holoway	kallistec: good question - coderanger?	9:14 am
coderanger	kallistec: It is designed to work alongside an oauth impl like oc-id while keeping the client/consumer dance out of erchef	9:14 am
holoway	coderanger: to put more words on that - basically oc-id would be the oauth2 endpoint, but it would hit this token interface behind the scenes?	9:15 am
coderanger	kallistec: oc-id handles the token creation UX but in the last step it requests a token from erchef and hands that back to the user so they can talk directly to erchef	9:15 am
	Yep	9:15 am
	<i>holoway high fives his brain</i>	9:15 am
jonlives	heh	9:15 am
coderanger	Means that if something better than oauth2 comes along, it is easier to plug in to the system	9:15 am
holoway	so this is one example of where I think it would be great for us to mark an RFC with stability	9:15 am
kallistec	coderanger: in terms of restrictions on characters in the token and other details, does oauth2 cover that stuff?	9:15 am
coderanger	(this could also support oauth1 if you wanted to)	9:15 am
holoway	because I think we should absolutely accept this, and we should mark it as unstable until after we actually learn what we really really mean	9:16 am
coderanger	kallistec: No, it has no guidance on that stuff. The places it covers does include the ways to pass tokens in though, and those are intentionally matched	9:16 am
holoway	(we could also use the word provisional, which is probably better than unstable)	9:17 am
kallistec	ok, then I think what I would want is one line of verbiage to say that it's intended to be compatible with oauth2 and oauth2 wins if there's a conflict	9:17 am
coderanger	Yeah, my biggest concern with accepting this is making sure someone is actually okay with implementing it	9:17 am

holoway	my guess is that we'll alter the RFC as we actually attack the problem	9:18 am
coderanger	Because I mostly wrote this up as a follow-up for the summit, not because I know how to implement it	9:18 am
holoway	which is my point - I would rather say "we want token auth, and this is a great stab at the spec"	9:18 am
	and mark that in the RFC	9:18 am
jmickle	agree this could be a tricky one to implement, and getting a decent management solution of the api tokens will be essential	9:18 am
	note that the ability to quickly pragmatically and through UI control tokens will be required	9:18 am
coderanger	holoway: I'm worried that we are very much diluting what an accepted RFC means if we treat it like that	9:19 am
jonlives	doesn't accepted / final make that distinction, or do we need another status?	9:19 am
holoway	coderanger: then we should have another status	9:19 am
coderanger	jonlives: Fair	9:19 am
holoway	mostly I see two things I'm worried about	9:19 am
tball	is a finished RFC supposed to be a full specification?	9:19 am
holoway	1) we're going to wind up doing big design up front	9:19 am
jonlives	tball: per https://github.com/opscode/chef-rfc/blob/master... , an accepted RFC is approved for implementation, a final rfc has been implemented.	9:20 am
	well, currently anyway.	9:20 am
ssd7	Do we have an idea of what the object capability system would like like. I worry that an initial implementation that doesn't have that in mind will make some decision that makes it hard to build in the future.	9:20 am

kallistec	If we like the term "rough concensus and working code" we have 1/2 of that.	9:20 am
holoway	2) as we actually attack the problem, we're going to force people back to the RFC process	9:20 am
	ya - so I think saying it is accepted but not final is real	9:20 am
coderanger	ssd7: The only thing specifically needed for objcap would be tokens restriced to a short TTL and only having a "whoami" permission	9:20 am
	holoway: I rescind my worries :)	9:21 am
	<i>holoway waves to the rescinded worries</i>	9:21 am
	holoway: We should probably loop back on marking stuff as final, I've not been good about that	9:21 am
jonlives	what about adding a "specification" track or something? ie, "this rfc is not the finished article, but an outline of an accepted proposal"	9:21 am
coderanger	holoway: Will look after mtg though	9:21 am
holoway	jonlives: I think you and coderanger should figure out if you feel like we need another word	9:21 am
jonlives	gotcha	9:21 am
holoway	I delegate completely to the two of you	9:22 am
	what we need to be able to say is	9:22 am
	+1 to token auth, this spec is a really, really, really good starting point	9:22 am
	as you encounter the issues while implementing, update the spec and circle back	9:22 am
coderanger	So for the immediate question, is there anything about this document currently that makes the server team sadpanda	9:22 am

	Or should we advance this now	9:23 am
holoway	sdelano: you around?	9:23 am
	he might already be heading to turkey	9:23 am
sdelano	I'm around, just hopped in, and haven't seen the RFC	9:23 am
	So it doesn't make me sadpanda yet :)	9:23 am
	<i>holoway grins</i>	9:23 am
coderanger	I know there were questions about global expires policies (which I'm super against), and some of the APIs for listing existing tokens	9:23 am
mary_grace	\]	9:23 am
holoway	coderanger: I think both of those can be mutated as we implement, for elegance and fit	9:24 am
	so - lets give sdelano time to read	9:24 am
	we'll circle back at the end of the meeting	9:24 am
	cwebber: don't let me forget	9:24 am
	next up - https://github.com/opscode/chef-rfc/pull/66	9:25 am
cwebber	holoway: :thumbs	9:25 am
holoway	root aliases in cookbooks, part of the festival de-noah	9:25 am
jonlives	feliz noah-dad!	9:25 am

clairem	\,.'	9:25 am
holoway	coderanger, tl-dr?	9:25 am
	jonlives: oh my god, that must be a meme	9:25 am
coderanger	tl;dr simplify the case of having one attributes file or one recipe file	9:25 am
holoway	can someone get a picture of noah with feliz noah-dad on it?	9:26 am
coderanger	because both of those are so common as to be worth not having a directory with one file in it	9:26 am
	So instead just a single file at the same level as metadata.rb (where the folders would be otherwise)	9:26 am
jonlives	holoway: your wish is my command: http://cdn.meme.am/instances/500x/56563325.jpg	9:28 am
coderanger	There seems to be general agreement on this for attributes given the almost complete dominance of the one-file form	9:28 am
tball	@jonlives	9:28 am
coderanger	recipes there was some dissent that this would be more confusing than it is worth	9:28 am
tball	@jonlives can you start with http://img1.wikia.nocookie.net/__cb201111242203... as a base?	9:28 am
holoway	I feel like there is little cost in making it work for recipes as well	9:29 am
	we can choose to *teach* however we want	9:29 am
coderanger	holoway: The cost would be the cognitive swap if we teach one format and then the other	9:29 am
jonlives	yeah - I'm for hitting attribs and recipes	9:29 am

coderanger	holoway: But I think that can be addressed	9:29 am
holoway	coderanger: I know, but you're going to have to soak that anyway; all the content in the universe uses the current form, so going to be a rough patch anyway	9:29 am
	lamont_oc talked about deprecating the attributes/* form, which I'm pretty against	9:30 am
	(simply because it costs nothing to continue to support it)	9:30 am
coderanger	Yeah, I don't think now is the time to deprecate anything	9:30 am
jonlives	agreed	9:30 am
lamont_oc	i don't even think now is the right time to deprecate that	9:30 am
holoway	if kallistec, lamont_oc, coderanger, jonlives all agree, that sounds pretty righteous	9:30 am
jonlives	+1 here	9:31 am
lamont_oc	i'd like to see a FC rule to favor the one-attribute-file	9:31 am
cwebber	30 MINUTE WARNING	9:31 am
lamont_oc	anyway i'm cool if everyone else is cool	9:31 am
coderanger	Okay, sounds like accept and move on	9:31 am
	holoway: you have final thumb	9:32 am
holoway	done	9:32 am
	ship that shit	9:32 am

	next up is https://github.com/opscode/chef-rfc/pull/57	9:32 am
	is mukta or siddheshwar here?	9:32 am
coderanger	holoway: This needs a copyright notice, I think you can probably authorize adding one if this was done as part of work for Chef Software	9:33 am
holoway	I think this is an easy accept. I'm also not sure we even need an RFC	9:33 am
	@coderanger yeah, we pay the bills	9:33 am
coderanger	holoway: Okay, do you place this document in the public domain yada yada? :)	9:34 am
holoway	yep. cwebber, can you take a note to have an agenda item to talk about what "size" of thingy needs an rfc?	9:34 am
cwebber	holoway: noted	9:35 am
holoway	or, to be more specific, we probably need to have LTs for subsystems start taking this on	9:35 am
coderanger	+1	9:35 am
holoway	ok - i accepted	9:35 am
	https://github.com/opscode/chef-rfc/pull/69	9:35 am
	audit mode	9:35 am
tball	I have the TL;DR	9:35 am
holoway	hit it	9:36 am
tball	The audit mode RFC has continued to generate comments and questions. The most common concerns are that:	9:36 am
	* The audit mode feature should be part of a plugin separate from core Chef	9:36

	Client. The counter to this is that chef developers want to be able to rapidly develop new core features and there is not a good plugin framework to leverage for that. The feature will also be shipped to consumer as part of an omnibus build, so the separation is primarily a developer concern, not a user concern.	9:36 am
	* Embedding Serverspec DSL into Recipe DSL seems unpopular because it adds complex DSL inside of complex DSL. But we are trying to flatten cookbook directory structure, and adding a new segment (named 'audit' folder) is contrary to this.	9:36 am
	** There is also concern that `controls` and `control` are too similar in the DSL.	9:36 am
	* The difference between the new audit phase and existing testing tools like ChefSpec and ServerSpec was highlighted. To recap, ChefSpec and ServerSpec are used for testing cookbook development. Audits are more similar to infrastructure monitoring - they will be ran regularly post-deployment to provide a stream of monitor-able data.	9:36 am
	** Audits are also similar to the existing minitest-chef-handler. They are a new phase of the chef run and as such can leverage additional logic, like being ran regardless of whether the converge was successful.	9:36 am
	The RFC was just updated. It now includes a `Rationale` section at its bottom which includes these common questions and the RFC maintainer's answers.	9:36 am
	We appreciate the continued feedback. I hope our updates help provide context for the ongoing discussion.	9:36 am
	my TL;DR probably needs a TL;DR	9:36 am
	<i>holoway laughs</i>	9:36 am
holoway	that was intense and prepared - thank you tball	9:36 am
someara	so... this is adding integration tests as another phase of a chef-client run?	9:37 am
coderanger	tldr this adds something like the chef-minitest-handle in a more formal way	9:37 am

jonlives	is there a separate RFC for the plugin stuff mentioned there? sounds like a pre-requisite for this.	9:37 am
coderanger	So you can run cross-component integration tests after converge	9:37 am
	jonlives: No, I meant cookbooks with lib files	9:37 am
	jonlives: Not a new plugins system	9:37 am
jonlives	ah ok cool	9:37 am
tball	@someara I think coderanger has the better TLDR if you're familiar with chef-minitest-handler	9:38 am
holoway	in particular, there is a use case for some folks to literally write the 'controls', and use that to verify state before remediation with a regular run	9:38 am
jmickle	is this already accomplished with dry run?	9:39 am
coderanger	I think we should also be clear this is being driven from business requirements that are super common at BigCos (often due to stuff like sox compliance)	9:39 am
	So the target is not small cloudy startups	9:39 am
holoway	jmickle: not really	9:39 am
kallistec	jmickle: no, there are certain negatives, like port such and such isn't open	9:39 am
someara	jmickle kinda... but you can't test "service delivers data on port 123" with dry-run	9:39 am
holoway	jmickle: the difference being the kinds of things you might look at tend to be side effects	9:39 am
coderanger	At least that is my read on it, it isn't stated super explicitly	9:39 am
kallistec	that don't map to chef primitives	9:39 am

lamont_oc	https://github.com/opscode/chef-rfc/pull/69#iss...	9:40 am
holoway	coderanger: for not bigco, someara has it right - this is essentially built in integration testing	9:40 am
cwebber	and in concept, you could have a cookbook that is maintained by auditor types exclusively with controls in it	9:40 am
holoway	(like ChefSpec is basically built-in unit testing via chefdk)	9:40 am
coderanger	holoway: Yes, just trying to help with any cognitive dissonance of "I wouldn't use this"	9:40 am
holoway	for sure	9:40 am
	and the room isn't full of bankers	9:40 am
	here to defend themselves :)	9:40 am
jonlives	i wouldn't use it, and I still think it's a good idea :p	9:40 am
jmickle	oh i suppose you could put this in cookbooks to validate after it runs?	9:41 am
holoway	jmickle: BINGO	9:41 am
	<i>holoway pulls the confetti down on jmickle</i>	9:41 am
coderanger	I agree it is a good idea, my concerns are mostly how we develop it and make sure we are happy with the DSL and usage thereof	9:41 am
holoway	for me, this is one that makes me want something between accepted and final	9:41 am
	I want to ship something, mark it as unstable, and not be bound by the laws of semver	9:41 am
jmickle	i agree this is a good idea, though i am concerned this will go horribly wrong with bad patterns	9:41 am

kallistec	jmickle: yes. The point is that you can run just the tests as part of an audit/compliance thing or you can use it as built-in integration tests	9:41 am
someara	I'm wondering how to map the suites to recipes	9:41 am
clairem	@jmickle there's a good set of comments on tests in cookbooks starting here https://github.com/opscode/chef-rfc/pull/69#iss...	9:42 am
jonlives	coderanger and I both feel that accepted is solid enough for this - most RFCs don't contain explicit implementation detail.	9:42 am
	accepted / final rather	9:42 am
jmickle	i think my concern is the same as coderanger's	9:42 am
holoway	I think we probably need to write up an RFC about a proposed release cycle and stability markers	9:42 am
	because I want to give this the thumbsup, and I also want to leave open the fact that we need to iterate quickly (for example, it feels wrongy-mc-wrong-pants to inline in recipes to me, but that might be because I haven't used the feature enough)	9:43 am
jmickle	holoway: +1	9:43 am
coderanger	I am enthusiastically +1 on having a system in core for audit events, capturing, displaying, etc. I think the initial development for the DSL should go in a cookbook and get merged to core later if there is consensus that there is a One True Way (or at least a small enough number of them to make core maint acceptable)	9:43 am
	It costs very little to keep the DSL out of core given how lib file loading works	9:44 am
someara	the cookbook could be iterated on way faster	9:44 am
jonlives	yeah	9:44 am
holoway	the fact that those are true statements makes me :(9:44 am
	managing all of that outside core will actually be super hard	9:45

		am
sersut	Are there any concerns for supporting rspec in addition to a DSL for controls?	9:45
		am
jonlives	i dunno, as an iteration pattern i think that's pretty cool	9:45
		am
	oh right, gotcha	9:45
		am
holoway	it's a new phase of the chef run	9:45
		am
	it'll be monkey-patch-a-palooza	9:45
		am
cwebber	15 MINUTE WARNING	9:45
		am
holoway	I would rather have it in core, have all of us have to deal with it	9:45
		am
	and be honest about its stability	9:45
		am
	(ie: it AINT STABLE YET)	9:45
		am
	essentially, feature flag it	9:45
		am
someara	rather than a partial search situation	9:46
		am
jmickle	holoway: if it goes into core though this could essentially slow down a chef-client run no?	9:46
		am
coderanger	holoway: We can add the hooks in core, the cookbook should just declare a function that gets passed the run context at the appropriate time	9:46
		am
holoway	someara: exactly	9:46
		am
sersut	jmickle: if the feature flag is off run performance will not be affected	9:46
		am
jonlives	+1 for feature flags.	9:46
		am

holoway	jmickle: it doesn't have to (as an implementaiton detail, easy enough o skip it)	9:46 am
coderanger	holoway: The plumbing seems relatively agreed upon, it is the DSL and how it gets written that is more contentious	9:46 am
holoway	this needs to be coupled with my other goal, which is I want a daily of the chef-client shipped asap	9:46 am
jmickle	holoway: fair enough, as long as it doesnt become a mandatory compile -> convergence step	9:46 am
	allow it to be hookable to be used	9:47 am
holoway	coderanger: right, which we can iterate on faster if its easy to try	9:47 am
	ie: no steps between you and the hotness (or the suck, as it may be)	9:47 am
coderanger	holoway: I still don't think people are going to be suuuuper comfortable installing dailies of something as omgcritical as chef	9:47 am
holoway	this is easy to test	9:47 am
	most people wont'	9:47 am
	but the folks in this room, or people who care about controls? they will :)	9:47 am
	and that's a perfectly good audience for me	9:48 am
jonlives	kind of a "them wot wants to can" deal, nothing wrong with still doing stable releases less frequently.	9:48 am
coderanger	Fair	9:48 am
holoway	you litreally cut stable releases	9:48 am
jmickle	i do agree that this should start out in a cookbook though	9:48 am

holoway	I really don[t think it can, from an internals point of view	9:48 am
	not and be clean	9:48 am
coderanger	It can't be only a cookbook, will require core changes first	9:48 am
	But the DSL itself can be a cookbook	9:48 am
holoway	we'll be evaluating the results through a straw	9:48 am
jmickle	coderanger: i think thats fair	9:48 am
coderanger	Basically when this launches, I want to be able to make a minitest version too	9:49 am
jonlives	I propose, given we're under 10 mins remaining, that discussion on that part moves to the PR cos it sounds like we're a ways from deciding	9:49 am
someara	can we just use serverspec	9:49 am
jmickle	holoway: just how i like everything... through a straw :-P	9:49 am
coderanger	If the APIs are flexible enough for that, I'm happy(er)	9:49 am
holoway	that is a way better statement htan "I want it in a cookbook"	9:49 am
	someara: I believe the current implementation uses the serverspec matcher	9:49 am
kallistec	coderanger: I want minitest support, too (or use bats or whatever you want)	9:49 am
jmickle	+1	9:49 am
kallistec	someara: yes, current stuff is serverspec	9:50 am

someara	woot	9:50 am
clairem	someara within controls you can write rspec3 with serverspec types and matchers	9:50 am
coderanger	holoway: Without a more modular approach a la cookbooks, I'm not sure how to support me writing something outside of core like that	9:50 am
jmickle	ok next meeting time for me...	9:50 am
holoway	coderanger: use feature flags and write it inside core, where it belongs	9:50 am
coderanger	(though, humorously, this is exactly the problem dialects are supposed to solve)	9:50 am
	holoway: That is terrible for people trying to develop a new option	9:51 am
kallistec	coderanger: the internal can be pluggable and we ship with a default which is serverspec	9:51 am
coderanger	holoway: rspec, minitest, bats, etc etc	9:51 am
holoway	coderanger: it doesnt have to be	9:51 am
coderanger	holoway: It means I either need commit bits or support from someone that has them	9:51 am
holoway	to do what? develop locally?	9:52 am
	pusha branch?	9:52 am
	you really don't	9:52 am
	anyway	9:52 am
coderanger	To release it in a way that others can use it	9:52 am

holoway	lets take it to the PR	9:52 am
coderanger	would mean building my own packages	9:52 am
cwebber	holoway: are we going to push on updates this week?	9:52 am
holoway	lets move to updates, and make sure the RFCs we haven't touched go to the top of the pile	9:53 am
	with this one at the bottom of the stack, in case it remains contentious :)	9:53 am
	cwebber: hit the updates	9:53 am
coderanger	holoway: Do we want to quickly approve the OSX one?	9:53 am
	holoway: You can probably accept that yourself	9:53 am
holoway	yeah	9:53 am
	lets do it	9:53 am
jonlives	+1	9:53 am
holoway	https://github.com/opscode/chef-rfc/pull/70	9:54 am
	<i>cwebber holds on updates</i>	9:54 am
	cwebber: go	9:54 am
coderanger	Do eeeeeet	9:54 am
cwebber	Community Update:	9:54 am

	CFP and Registration for ChefConf is now open	9:55 am
	it will be March 31 - Apr 2nd at the Santa Clara Convention Center	9:55 am
	Tickets for contributors are \$700 with discount code CHEFSTAR	9:55 am
	The plan is to shutdown the wiki on 1 January: http://lists.opscode.com/sympa/arc/chef/2014-11...	9:56 am
	A Chef Community on StackExchange has been proposed - http://area51.stackexchange.com/proposals/77609...	9:56 am
coderanger	(reminder: if you are watching the stackexchange thingy make sure to vote up example questions)	9:57 am
cwebber	There is an Awesome kickstarter project in progress from our very own coderanger - https://www.kickstarter.com/projects/coderanger...	9:57 am
	go check it out and support it	9:57 am
github-bawt	[chef-rfc] danielsdeleo created ruby-193-eol (+1 new commit): http://git.io/ZKHCPA	9:57 am
	chef-rfc/ruby-193-eol 67d9e4d danielsdeleo: Propose preemptive EOL of Ruby 1.9.3	9:57 am
cwebber	And that is it for community	9:57 am
coderanger	(shoutout to holoway for Chef Software backing the campaign)	9:58 am
holoway	sersut!	9:58 am
sersut	On the client release front we have the final RC for 12.0.0 out. Unless any new blockers come up we're planning to release 12.0.0 next Monday.	9:58 am
	Also 12.2.0.alpha.0 is out as an experimental release if you want to play with audit mode and see how it feels.	9:58 am
	that's for Client side..	9:59 am

holoway	any other business?	9:59 am
jonlives	holoway: on the chef software note, I'd like to quickly check what needs to happen for https://github.com/opscode/chef/pull/2423 to be merged?	9:59 am
kallistec	BTW folks, have been meaning to discuss http://git.io/ZKHCPA	9:59 am
jonlives	re RFC-030	9:59 am
sdelano	Server Update: We've had tons of great feedback from the Chef 12 RCs over the past couple months and yesterday we finished up the last of our internal testing based on the revisions that have been made during the RC process. We're currently in the process of pushing the Chef 12 GA release out the door. Expect it to hit the blogs / mailing list in the next couple hours.	10:00 am
holoway	jonlives: I just need to tweak it	10:00 am
jonlives	gotcha	10:00 am
holoway	sdelano: woot!	10:00 am
	kallistec: have cwebber put that on the next agenda	10:00 am
coderanger	sdelano: Are the issues with the config file fixed?	10:00 am
sdelano	Sorry, I snuck that one in there after the shot-clock expired	10:00 am
coderanger	sdelano: They weren't in the last RC	10:00 am
kallistec	holoway: I think we need to expedite, if 12.0 is going out on monday	10:01 am
holoway	kallistec: then the answer is no	10:01 am

kallistec	Proposal matches what's already been accepted in RFC 015	10:01 am
holoway	I see	10:02 am
	I wish ranjib was here	10:02 am
kallistec	but there was some hesitation to follow through on it	10:02 am
sdelano	coderanger: let's chat. Not recalling the specifics of the issue, there were a lot of config changes.	10:02 am
holoway	kallistec: it feels in line with what we said we would do	10:02 am
	kallistec: but in my heart, I know we're going to bone someone for essentially no reason	10:03 am
	ie: it's gonna work on 1.9	10:03 am
kallistec	holoway: yes, we will, the question is if we do it at our 12.0 or wait for ruby-core to bone them in 3 months	10:04 am
holoway	we can drop support for it without hard depending it	10:04 am
ssd7	Given the length of time that Chef 12 might be supported for, I'm not sure I would call it no reason. Unless we think we can drop support midway through the lifecycle	10:04 am
holoway	we must drop support	10:04 am
	so thats easy	10:04 am
	the question is, do we set the hard dependency, or let people YOLO	10:04 am
cwebber	holoway: call the meeting and continue discussion?	10:05 am
jonlives	I'd say at the least there should be a big old deprecation warning.	10:05 am

	during a run	10:05 am
holoway	cwebber: yes	10:05 am
lamont_oc	to drop support we want to rip out travis testers, we may then start writing >= 2.0 only code, and the client may randomly fail when you hit those codepaths	10:05 am
holoway	thats okay	10:05 am
cwebber	— MEETING END --	10:05 am