

# SATURN 2015

11th Annual SEI Architecture Technology User Network Conference

April 27–30, 2015 | Baltimore, MD | [www.sei.cmu.edu/saturn/2015](http://www.sei.cmu.edu/saturn/2015)



## SATURN 2015 Call for Submissions

Submissions for the Software Engineering Institute (SEI) Architecture Technology User Network (SATURN) Conference 2015 technical program are now being accepted. Please submit proposals for 15-, 30-, and 90-minute sessions to the online submission system no later than January 16, 2015. Proposals will be reviewed and accepted monthly. Proposals not accepted early will receive feedback from a committee of software-architecture community leaders that can be used to refine the proposal before the final deadline.

Submitters who are accepted to give 90-minute presentations will receive free conference registration and one free night in the conference hotel. Those accepted to give 15- and 30-minute presentations will receive 60% off the full conference-registration fee. This compensation applies to one speaker in a multiple-author talk.

## What's new for 2015

**We have three big changes in our format that we are excited to tell you about:**

1. The main conference program will run for three full days.
2. Sessions will be of different lengths (15, 30, or 90 minutes).
3. Shorter tutorials will now be included in the main conference program.

We are also expanding our rolling-acceptance process from last year, so submit early to get feedback and improve your chances.

## Conference themes

This year's technical program will cover a variety of subjects organized into three themes: (1) Technology, (2) Methods and Tools, and (3) Leadership and Business. A session might touch on more than one of these themes, but successful sessions will have a primary theme that will be the main focus.

### Technology

This theme comprises topics related to technology in the context of software architecture. Specific examples that explore technical details and stories (success or failure) that demonstrate how system properties are promoted or inhibited by technology choices are a good fit. Diagrams and code are welcomed.

Suggested topics include, but are not limited to, the following:

- Choosing and using back-end or front-end technologies and frameworks. For example,
  - Web—Ruby on Rails, ASP.Net, Sinatra, GWT, Django, CakePHP, Express, modern JavaScript frameworks (such as Ember.js, Angular.js, jQuery, etc.), MEAN stack, LAMP stack, etc.

- Mobile—Swift, Android, Cocoa, Windows Modern/Metro, Apache Cordova, etc.
- Server/back ends—J2EE, .Net, JRuby, Node.js, JBoss, etc.
- Data management—Hibernate, Spring, ADO.Net, LINQ, Object Relational Mappers, etc.
- Cloud-computing architectures, for example, success and failures for systems architected and deployed to Heroku, Azure, AWS, Bluemix, Google AppEngine, CloudFoundry, OpenBees, OpenShift, etc.
- Architecting for modern information-management systems and computing styles—cognitive systems, big data, NoSQL, MapReduce, stream-based computing, information discovery, indexing/search, or data warehousing
- Architecting for DevOps—patterns, practices, and technologies for promoting DevOps practices such as monitoring, fault prediction, managing parallel feature branches for phased rollouts or A/B testing, virtualization, and tools such as Chef, Docker, Vagrant, Puppet, etc.
- Architecting for specific engineering paradigms including
  - Virtualization, multi-tenant, cloud, or stand-alone/dedicated systems
  - Embedded or real-time systems, including Arduino and Raspberry Pi
  - Legacy systems
  - Service-oriented architectures (SOAs) and concurrent or event-driven systems
  - High-scale, high-volume systems such as web applications, enterprise resource planning (ERP) packages, and control systems

### Methods and Tools

This theme comprises topics related to experiences tailoring and using software architecture analysis, design, and evaluation methods and supporting tools. Innovative new methods and tools are just as important as variants and lessons from the field using existing methods and tools. Anecdotes and evidence-based experiences are welcome.

Suggested topics include, but are not limited to, the following:

- Combining methods from different communities: for example, object-oriented analysis and design techniques, lean and agile practices, and architectural methods and notations

- Enforcing architectural decisions through architecturally evident coding styles, component interface design principles, or architectural patterns
- Managing software evolution and technical debt, enterprise architecture frameworks, and design for operations
- Architecting in uncertainty, delaying decisions while remaining effective, or dealing with volatile and quickly changing requirements

## Leadership and Business

This theme comprises topics related to the software architect's role in leading teams and influencing business decisions. Topics relevant to both organizations and professionals that address the human elements of software architecture are a good fit.

Suggested topics include, but are not limited to, the following:

- Leveraging architecture for project management activities such as estimation, planning, and tracking; planning agile pre-games or conducting architectural spikes and sprint zeros; managing technical risk; and supporting make-or-buy decisions
- Providing architectural coaching, establishing communities of practice, and offering architecture certification, training, and education
- Succeeding in hard-sell situations: for example, risk identification or assessing business value of refactoring and legacy-system modernization
- Adopting agile collaboration practices when working with stakeholders
- Software development on self-organizing teams including highly agile teams or open source projects; enabling effective architectural refactoring and evolution
- Developing leaders, teams, and organizations that are architecturally aware “systems thinkers” prepared to lead and build the next wave of innovative software systems

## Session types

When deciding what kind of session to propose, choose a session type that you are comfortable presenting or facilitating and that also helps you to effectively share your wisdom, knowledge, and experience with your audience.

- Experience-Report Presentation—15- or 30-minute presentation that describes a first-hand experience, lessons learned, or practitioner-focused research. Generally a slide-based presentation.
- Participatory Session—90-minute facilitated workshop with at least 50% of the time dedicated to hands-on, active learning activities with attendees, such as writing code, applying a design technique, or practicing an architecture leadership activity. Please tune the activities to work with groups of 10 to 50 people.
- Tech Talk or Tutorial—30- or 90-minute session that is lecture-based and focused on teaching specific topics in depth. Look for opportunities to package longer content as multiple independent sessions rather than parts that presume former sessions as prerequisites.

## Submission

All proposals must be submitted to the online submission system no later than January 16, 2015. Proposals will be reviewed and accepted monthly. Proposals not accepted early will receive feedback from a committee of software-architecture community leaders that can be used to refine the proposal before the final deadline.

For tips on how to submit a winning proposal, see “SATURN 2015 Proposal Guidelines” on the SATURN blog at <http://saturnnetwork.wordpress.com/2014/09/15/saturn-2015-proposal-guidelines/> and also see “7 Tips for Writing a Great Submission Proposal” at <http://saturnnetwork.wordpress.com/2013/12/12/7-tips-for-writing-a-great-submission-proposal-for-saturn-2014/>.

To help ensure your proposal is clear and gets a fair hearing, please include all of the following information, which can be found in the template at <http://www.sei.cmu.edu/saturn/2015/call-for-submissions> and uploaded as a PDF file with your abstract in the online submission system:

- Title for your session
- Abstract (250 words or less)—Think “elevator pitch.” Spill the beans: summarize the core ideas of your session (“... our experiences shows that mixing X and Y helps build scalable systems”) instead of making vague promises (“... discusses three interesting topics about scalability”).
- Session theme—Pick one of {Technology, Methods and Tools, Leadership and Business}
- Session type—Pick one of {experience-report presentation, participatory session, or tech talk/tutorial}
- Session length—Choose one (or more) of 15, 30, or 90 minutes. Every year we must reject talks because there's only so much time in the schedule, so please let us know your preferred length as well as shorter times if they can work, as this gives us more schedule flexibility.
- Session mechanics—How will the session be conducted? Should include information such as an outline, timing, and description of activities (for participatory sessions). Please limit the description to no more than 700 words. If you already have blog posts, videos, or other materials for this session, feel free to include links.
- Learning outcomes—What will attendees learn and be able to apply with their teams after the conference?
- Target audience and assumed knowledge of the audience
- Names and bios for all speakers. Bios should be 100 words or fewer.
- Has this content been presented before? Please list.
- Speaking experience—Have you presented at previous conferences? Please describe.

You may submit the same or similar topics to multiple session types; however, please note this in your proposals and tailor your proposals accordingly.

Good luck with your submission! We're looking forward to seeing the great talks you propose!