CFP abstract guide

Manaswini Das

Title of your talk: Be Concise. Be Clear. Be Clean

- Try to answer those questions:
  - **WHAT** is this about?
  - **WHO** is this for?
  - **FEEL** What they FEEL right now
  - **OUTCOME:** Why should they attend it? What are you PROMISING them to learn, what they get in exchange?
  - **HOW MANY?** How many things you will talk about during your talk?
- Don’t use the same title and abstract for every conference
- Make the title solution-focused, actionable.
- Length: A good rule of thumb is to have a title of 12 words or less.

Writing abstract: Be Concise. Be Clear. Be Clean

To begin writing an abstract, we need to learn what the topic of proposal is going to be, who'd be our audience and then set expectations to ensure they know what to expect.

The abstract should include the following informations:

- Who is this talk for?
- Why should they be interested?
- What information will be covered and the audience will learn from the session?
- How will it be explained/demonstrated?
- Any background information/knowledge the audience needs in order to participate or understand your presentation?
- Consider including a call-to-action or next steps for your audience

Once it's written..

Revise your abstract, edit it properly, and ask for feedback from your colleagues / friends. It is important that the language is inclusive and persuasive, and the information concise.

Some sample accepted abstracts from my end:

GitHub Universe 2021:
Title: Dependency management using git
This talk will cover how git can aid in dependency management in-house by using submodules, subtrees, and subrepo. Manaswini will also review the caveats, differences, applications—for example, single-sourcing documentation and publishing the same documentation in various platforms such as wiki and gitbook—and other alternatives, like git repo and gitslave, that can help you achieve dependency management.

AWS Shebuilds Summit 2021:
Title: The case for REST
GraphQL, the query language for graph databases, is gaining traction mainly due to its ease of use and how it addresses over-fetching and under-fetching. Every API architectural style has its own pros and cons, applications, and caveats. There is no use steering clear of one for a simple use case when you can incorporate it on your own in the existing style. While REST is preferred for applications involving CRUD, GraphQL is preferred for building composite APIs/microservices. This article will walk you through over-fetching and under-fetching without shifting to an entirely new architectural style like GraphQL. We can use sparse fieldsets or field grouping to handle over-fetching, and we can use compound documents for under-fetching.

Dos and don’ts

Dos:

- Be specific and to the point - use plain English without using jargon or unnecessary detail
- Include live demos, examples, stories, implementation tools, etc.
- Keep in mind that abstract and title should help your audience (reviewers and attendees) decide if they want to attend or not
- Stay within the word count on the CFP - A good rule of thumb is two to three paragraphs clearly expressing your idea.
- Study examples from last year's CFP submissions for the event

Don’ts:

- Don’t use the same abstract for all the conferences.
- Don’t be too wordy or share a link as a replacement for abstract.
- Don’t write vague abstracts or titles - attendees won’t attend a talk they don’t know what it is about
- Avoid abbreviations or cliche.
- Don’t leave CFP until the last minute

Happy writing and presenting! :)

Thank you