Why Attend ArchConf

ArchConf is an educational event for software architects, technical leaders, and senior developers. Presented to you by No Fluff Just Stuff. No Fluff Just Stuff is focused on delivering a completely attendee-centric experience, free from vendor bias.

WE OFFER

LEARN FROM THE BEST SPEAKERS
Our speakers are not simply vendor representatives — they are industry recognized experts. They are published authors, consultants, executives, and open source leaders.

IN-DEPTH 90 MINUTE SESSIONS
Our longer session format, workshops, and multi-part sessions allow speakers to go in-depth and teach the detailed concepts you need to know.

INCREASE EFFICIENCY AND AGILITY
Technology changes, it’s a fact of life! And while many developers are attracted to the challenge of change, many organizations do a particularly poor job of adapting. ArchConf will explore ways to achieve competitive advantage with new technology.

CLOUD ARCHITECTURES
ArchConf will explore different cloud computing architectures and how you can take advantage of them.

NO VENDOR HALL
NFJS understands that the attendee is the customer, not the legions of sponsors found at every other conference. ArchConf is an environment for growth & learning, not vendor sales pitches & sponsored sessions.

DEVELOP YOUR SOFT SKILLS
Effective architects need significant technical depth and breadth and domain knowledge. In addition, there is another perhaps more vital aspect of being an architect - the soft skills. Communication, leadership, persuasion and more.

CONTINUOUS DELIVERY
As systems get more complex, it takes longer to deliver customer requirements. To meet customer demands, organizations must embrace automated testing, continuous integration, and continuous delivery.

Registration is open! Register today @ www.ArchConf.com
Monday, Dec. 9

8:00 - 9:00 AM: EARLY REGISTRATION: PRE-CONFERENCE WORKSHOP ATTENDEES ONLY - OPAL BALLROOM FOYER

9:00 - 6:00 PM: Sessions

Session #1: Architecture: The Hard Parts by Neal Ford and Mark Richards
Architects often look harried and worried because they have no clean, easy decisions: everything is a terrible tradeoff. Architecture has lots of difficult problems, which this workshop highlights by investigating what makes architecture so hard. In this hand-on platform-agnostic architecture workshop we go beyond the fundamentals of software architecture and focus on the really hard problems. We'll focus on areas surrounding modularity verses granularity, the challenges of event-driven architectures (including difficult error handing issues), why reuse doesn't work anymore, how to do tradeoff analysis, and how to decouple services to achieve proper granularity. Architecture is full of hard parts; by tracing the common reasons and applying lessons more universally, we can make it softer.

Session #2: Applying Design Patterns by Venkat Subramaniam
Learning about design patterns is not really hard. Using design patterns is also not that hard. But, using the right design pattern for the right problem is not that easy. If instead of looking for a pattern to use if we decide to look for the design force behind a problem it may lead to better solutions. Furthermore, with most mainstream languages supporting lambda expressions and functional style, the patterns appear in so many more elegant ways as well.

Session #3: Hands-On experience with Data as a Service by David Sietz
Should Information Management systems apply the services architecture? Many data provisioning and BI systems are monolithic, tightly coupled, difficult to scale, and stumble when it comes to delivering MVP in a timely manner. Data as a Service delivers MVP in real-time data management, while avoiding many of the ant-patterns that traditional data provisioning and BI systems portray. Unlike traditional BI tooling, building out a Data as a Service system doesn't require high up-front costs and the welding of multiple products. Get hands-on experience learning how the Rust language, a Kafka broker, and CouchDB cluster can be used to build out a DaaS system that delivers faster and more scalable solutions to your customer.

Session #4: Agility Through Modular Architectures - From Modular Monoliths to Microservices by Kirk Knornischl
No single architectural style solves all needs. Though microservices have taken the developer community by storm recently, they are not always the optimal solution. In some cases, a more monolithic architecture may be more suitable short term. Or perhaps a more traditional system of web services that allow you to leverage existing infrastructure investment is preferable. Fortunately, proven architectural practices allow you to build software that transcends specific architectural alternatives and develop a software system that gives the development team the agility to shift between different architectural styles without undergoing a time-consuming, costly, and resource intensive refactoring effort. Modularity is the cornerstone of these alternatives.

Session #5: Developer To Architect by Nathaniel Schutta
Becoming a software architect is a longed-for career upgrade for many software developers. While the job title suggests a workday focused on technical decision-making, the reality is quite different. In this workshop, software architect Nathaniel Schutta constructs a real-world job description in which communication trumps coding.

Session #6: Kubernetes Working for You by Jonathan Johnson
At the end of this workshop, you will be comfortable with designing, deploying, managing, monitoring and updating a coordinated set of applications running on Kubernetes. Distributed application architectures are hard. Building containers and designing microservices to work and coordinate together across a network is complex. Given limitations on resources, failing networks, defective software, and fluctuating traffic you need an orchestrator to handle these variants. Kubernetes is designed to handle these complexities, so you do not have to. It’s essentially a distributed operating system across your data center. You give Kubernetes containers and it will ensure they remain available. Kubernetes continues to gain momentum and is quickly becoming the preferred way to deploy applications.

In this workshop, we’ll grasp the essence of Kubernetes as an application container manager, learning the concepts of deploying, pods, services, ingress, volumes, secrets, and monitoring. We’ll look at how simple containers are quickly started using a declarative syntax. We’ll build on this with a coordinated cluster of containers to make an application. Next, we will learn how Helm is used for managing more complex collections of containers. See how your application containers can find and communicate directly or use a message broker for exchanging data. We will play chaos monkey and mess with some vital services and observe how Kubernetes self-heals back to the expected state. Finally, we will observe performance metrics and see how nodes and containers are scaled. Come to this workshop the learn how to deploy and manage your containerized application. On the way, you will see how Kubernetes effectively schedules your application across its resources.

Session #7: Machine Learning Workshop by Brian Sletten
Machine Learning is all the rage, but many developers have no idea what it is, what they can expect from it or how to start to get into this huge and rapidly-changing field. The ideas draw from the fields of Artificial Intelligence, Numerical Analysis, Statistics and more. These days, you’ll generally have to be a CUDA-wielding Python developer to boot. This workshop will gently introduce you to the ideas and tools, show you several working examples and help you build a plan to for diving deeper into this exciting new field.

ArchConf
-Session Schedule-
(event schedule as of September 18, 2019)
Session #8: Analyzing Architecture by Alexander von Zitzewitz
Analyzing architecture is all about finding structural decay in applications and systems to determine whether the architecture is still satisfying the business concerns (performance, scalability, fault tolerance, availability, and so on) and also whether the architecture supporting the application functionality is still viable. This is known as “architectural vitality”. While the functionality of a system may be sound, the architecture supporting that functionality may not be. For example, performance and scalability may have been the number one concern 5 years ago, but today agility, testability, and deployability is the number one concern to support high levels of competitive advantage and time-to-market. Does the architecture support these "-ilities"? If not, the company is likely to fail in today's highly competitive market.

5:00 - 6:30 PM: MAIN ARCHCONF REGISTRATION: OPAL BALLROOM FOYER
6:30 - 7:30 PM: OPENING NIGHT DINNER: OPAL BALLROOM
7:30 - 8:30 PM: ARCHCONF 2019: OPENING NIGHT KEYNOTE
8:30 - 10:30 PM: OPENING NIGHT OUTDOOR RECEPTION: POOL LAWN

Tuesday, Dec. 10

7:30 - 8:30 AM: BREAKFAST & LATE REGISTRATION: OPAL BALLROOM
8:30 AM - 10:00 AM: Sessions

Session #9: Microservices Migration Patterns by Mark Richards
The path to migrating to Microservices from a monolithic or service-oriented architecture (or even starting a greenfield application) is riddled with challenges, pitfalls, canyons, demons, and even fire-breathing dragons. I like to call it "The Kings Road". In this session I will show the migration patterns that allow you to easily fly over this challenging road and ease the pain associated with moving to microservices. I will also show you some automation tools you can use to help analyze your applications to determine how challenging this road will be.

Session #10: Designing for Resilience and Scale by Venkat Subramaniam
Why talk about resilience when thinking of scale? It turns out all the effort we put in to achieve great performance may be lost if we're not careful with failures. Failure is not only about unavailability of parts of an application to some users, it may result in overall poor performance for everyone else as well.

Session #11: From Monolith to 20,000 Deployments by Chris Maki
A down in the trenches look at building, running and day-to-day development with a Continuous Delivery pipeline. This talk is based on my experiences building multiple CD pipelines and optimizing developer workflows to push changes to production all day. I'll walk you through how we transformed a two-day deployment process into a 20-minute CD pipeline and then go on to perform more than 20,000 deployments.

Session #12: Fixing Software Architecture by Kirk Knoernschild
Current approaches to software architecture do not work. As we challenge some of the sacred truths of software development (reuse, failure prevention), we examine how current approaches to software architecture must also change.

Session #13: Consume First Architecture by Llewellyn Falco
Let's get back to basics. One of the microskills often used in TDD is Consume First Architecture, which simply means using the fields and methods "before" they exist. Sounds easy? Well yes and no. Even simple lines of code can have HUGE implications on your architecture. The real skill in consume first is to be able to see, question and respond to those implications on sight.

Session #14: Event-driven Microservices by Jeremy Deane
Using the Microservices Architectural Style to incrementally adopt an Event-driven Architecture (EDA) lowers up-front costs while decreasing time-to-market. EDA extracts value from existing occurrences, limiting invasive refactoring or disrupting existing application development efforts. Implementing Event-driven Microservices yields intelligence, scalable, extensible, reactive endpoints.

Session #15: Architecture Foundations: Characteristics & Tradeoffs by Neal Ford
This session describes how architects can identify architectural characteristics from a variety of sources, how to distinguish architectural characteristics from domain requirements, and how to build protection mechanisms around key characteristics. This session also describe a variety of tradeoff analysis techniques for architects, to try to best balance all the competing concerns on software projects.

Session #16: The Ultimate Metric by Arty Starr
Since the dawn of software development, we've struggled with a huge disconnect between the management world and the engineering world. We try to explain our problems in terms of "technical debt", but somehow the message seems to get lost in translation, and we
drive our projects into the ground, over and over again. What if we could detect the earliest indicators of a project going off the rails, and had data to convince management to take action? What if we could bridge this communication gap once and for all?

10:00 - 10:30 AM : MORNING BREAK: OPAL BALLROOM FOYER

10:30 - 12:00 PM - Sessions

Session #17 : Microservices and Distributed Data by Mark Richards
Jorge Santayana is famous for saying "Those who cannot remember the past are condemned to repeat it". When SOA (Service-Oriented Architecture) was all the craze, everyone got all excited about services, but forgot about the data. This ended in disaster. History repeats itself, and here we are with Microservices, where everyone is all excited about services, but once again, forgets all about the data. In this session I will discuss some of the challenges associated with breaking apart monolithic databases, and then show the techniques for effectively creating data domains and how to split apart a database. I consider the data part of Microservices the hardest aspect of this architecture style. In the end, it's all about the data.

Session #18 : The Future of Threads on the JVM: Impact of Continuations and Fibers by Venkat Subramaniam
Once considered lightweight, threads in reality take up significant memory and thus turn into a limitation for true scale. The JVM is heading towards creating fibers which are lightweight compared to threads and have the potential to be truly non-blocking. Mixed with continuations, which are data structures that can preserve state between calls, we can create highly effective asynchronous applications that can scale to a much greater extent than threads.

Session #19 : Cloud Native Applications by Chris Maki
How do you build a Cloud Native Applications? So many cloud deployments are a lift and shift architecture, what would it look like if you started from scratch, only used cloud native technologies? During this session we will compare and contrast two applications, one built using a traditional Java application architecture, the other using a cloud native approach. How does building an app for the cloud change your architecture, application design, development and testing processes? We'll look at all this and more.

Session #20 : Modularity: The Foundation of Modern Architectures and Platforms by Kirk Knoernschild
Modularity is the common aspect of modern architectures and platforms. Understanding the role of modularity when making architecture decisions is critical.

Session #21 : Developing Design Sense for your Code by Llewellyn Falco
“In order to make delicious food… you need to develop a palate capable of discerning good and bad. Without good taste, you can’t make good food.” - Jiro Ono (World’s Best Sushi Chef) Many of us are stuck with messy code. We know it’s not great but it works and what can we do? Where and how do you start?

Session #22 : AMQP Messaging Fundamentals by Jeremy Deane
This two session workshop covers AMQP messaging concepts and technologies including hands-on exercises with RabbitMQ, Spring and Docker

Session #23 : Architecture Foundations: Styles & Patterns by Neal Ford
This session covers basic application and distributed architectural styles, analyzed along several dimensions (type of partitioning, families of architectural characteristics, and so on).

Session #24 : Data-Driven Retros in Practice by Arty Starr
How does your team decide what's the most important problem to solve? When we ask a question like "what's the biggest problem?", it doesn't mean the biggest problems will come to mind. Instead, we're biased to think about what's bothered us most recently, annoyances, or pet peeves. It's really easy to spend tons of time working on improvements that make little difference. But what if we had data that pointed us to the biggest problems across the team?

12:00 - 1:30 PM : LUNCH: OPAL BALLROOM

1:30 - 3:00 PM - Sessions

Session #25 : Microservices Caching Strategies by Mark Richards
Have you ever wondered how to share data between microservices? Have you ever wondered how to share a single database schema between hundreds (or even thousands) of microservices (cloud or on-prem)? Have you ever wondered how to version relational database changes when sharing data in a microservices environment? If any of these questions intrigue you, then you should come to this session. In this session I will describe and demonstrate various caching strategies and patterns that you can use in Microservices to significantly increase performance, manage common data in a highly distributed architecture, and even manage data synchronization from cloud-based microservices. I'll describe the differences between a distributed and replicated cache, Using live coding and demos using Hazelcast and Apache Ignite, I'll demonstrate how to share data and also how to do space-based microservices, leveraging caching to its fullest extent.
Session #26: GraalVM—What and How it will impact your technical decisions by Venkat Subramaniam
GraalVM is a polyglot environment that can execute your code, written in multiple languages, in multiple different platforms. With version 1.0 released this year, this technology has the potential to make significant impact on both development and deployment.

Session #27: Serverless by Chris Maki
I hope you'll join me on this exciting survey of Serverless Computing. When you think of Serverless you probably think of Lambda's or Cloud Functions but there's so much more to the Serverless ecosystem. During this session will look at Serverless Computing in all its various forms and discuss why you might want to use a Serverless architecture and how it compares to other cloud services.

Session #28: Migrating to the Java Platform Module System by Kirk Knoernschild
The Java Platform Module System was available with Java 9. In this session, we provide a clear framework for migrating your applications to JPMS.

Session #29: Cutting Code Quickly by Llewellyn Falco
In this guided demo, we are going to look at 3 different techniques that are remarkably powerful in combination to cut through legacy code without having to go through the bother of reading or understanding it. The techniques are: Combination Testing: to get 100% test coverage quickly, Code Coverage as guidance: to help us make decisions about inputs and deletion, Provable Refactorings: to help us change code without having to worry about it. In combination, these 3 techniques can quickly make impossible tasks trivial.

Session #30: AMQP Messaging Fundamentals (continued) by Jeremy Deane
This two session workshop covers AMQP messaging concepts and technologies including hands-on exercises with RabbitMQ, Spring and Docker

Session #31: Restructuring & Migrating Architectures by Neal Ford
Patterns/antipatterns, techniques, engineering practices, and other details showing how to restructure existing architectures and migrate from one architecture style to another.

Session #32: Anatomy of Culture by Arty Starr
What makes software development complex isn't the code, it's the humans. The most effective way to improve our capabilities as an organization is to better understand ourselves. In this session, we'll breakdown the dynamics of culture into explicit architecture models based on a synthesis of research that spans cognitive science, biology and philosophy. We'll discuss the nature of Identity, communication, relationships, leadership and human motivation by thinking about humans like code!

3:00 - 3:15 PM: BREAK: OPAL BALLROOM FOYER
3:15 - 4:45 PM - Sessions

Session #33: The Rise and Fall of Microservices by Mark Richards
In 250BC Rome began its expansion into Carthage, and later into the divided kingdoms of Alexander, starting the rise of a great empire until its decline starting around 350AD. Much can be learned from the rise and fall of the Roman Empire as it relates to a similar rise and fall: Microservices. Wait. Did I say “fall of microservices”? Over the past 5+ years Microservices has been on the forefront of most books, articles, and company initiatives. While some companies been experiencing success with microservices, most companies have been experiencing pain, cost overruns, and failed initiatives trying to design and implement this incredibly complex architecture style. In this session I discuss and demonstrate why microservices is so vitally important to businesses, and also why companies are starting to question whether microservices is the right solution. Sir Issac Newton once quoted “What goes up must come down”; Blood, Sweat & Tears sang about this in their hit "Spinning Wheel". Microservices is no exception. Come to this provocative session to learn about the real challenges and issues associated with microservices, how we might be able to overcome some of the technical (and business) challenges, and whether microservices is really the answer to our problems.

Session #34: Tools for Big Data Processing: A Look at the Landscape by Venkat Subramaniam
Big data comes in two flavors high volume and high frequency. How we process the data depends on both the nature of data and the type of applications.

Session #35: Docker to the Rescue by Chris Maki
Docker has revolutionized how we build and deploy applications. While Docker has revolutionized production, it's also had a huge impact on developer productivity. Anyone that's used Docker for an extensive period of time will tell you it's a blessing and a curse. Yes, it's portable but networking and other characteristics of Docker can make the most chill developer long for plain old Java. During this session we'll look at Docker's good points and its painful ones. The end goal - enable anyone on your team to go from zero to productive in under 20 minutes.

Session #36: Java MicroProfile - Enterprise Java Meets Microservices by Kirk Knoernschild
Organizations have a lot of expertise in Java EE. With MicroProfile, developers can leverage this expertise to build cloud-native applications.
Session #37 : Mob Programming by Llewellyn Falco
If you think pairing programming (2 people on 1 computer) is crazy, hold onto your hats; it’s time for Mob Programming. Mob Programming: All the brilliant people working on the same thing, at the same time, in the same place, and on the same computer.

Session #38 : Architectural Resiliency by Jeremy Deane
Regardless of the techniques used to make an enterprise solution Highly Available (HA), failure at some point inevitable. Resiliency is how fast a system reacts and then recovers to such failures. This session will cover a number of techniques and patterns such as Intelligent Agents, Tolerant Reader, and Circuit Breaker, for addressing architectural resiliency.

Session #39 : Automating Architecture Governance Using Fitness Functions by Neal Ford
This session describes mechanisms to automate architectural governance at application, integration, and enterprise levels.

Session #40 : Anatomy of Communication by Arty Starr
In its essence, DDD (Domain-Driven Design) is a philosophy and set of techniques for constructing a shared mental model of a system, and translating it into software. Ideally, we want conversations with other humans, and conversations with our code, to have the highest possible bandwidth. Every major problem in software development boils down to communication. What if we enabled a dialog around the challenges, by first deconstructing the Anatomy of Communication itself using our same DDD techniques?

4:45 - 5:00 PM : BREAK: OPAL BALLROOM FOYER

5:00 - 6:30 PM - Sessions

Session #41 : 6 Essential Architecture Practices by Mark Richards
Software architecture is hard. It is full of tradeoff analysis, decision making, technical expertise, and leadership, making it more of an art than a science. The common answer to any architecture-related question is “it depends”. To that end, I firmly believe there are no “best practices” in software architecture because every situation is different, which is why I titled this talk “Essential Practices”: those practices companies and architects are using to achieve success in architecture. In this session I explore in detail the top 6 essential software architectural practices (both technical architecture and process-related practices) that will make you an effective and successful software architect.

Session #42 : Seven Technologies to Keep an Eye On by Venkat Subramaniam
The world we live in today is changing rapidly, both in terms of hardware and in business demands.

Session #43 : Managing teams in chaos by Chris Maki
The cloud promises highly scalable infrastructure, economies of scale, lower costs and a more secure platform. When moving to the cloud, how do you take advantage of these new capabilities? How do you optimize your organization to make the best use of the resiliency and elasticity offered by the cloud? Closely associated with cloud computing is Continuous Delivery, the automated process to get changes to your customers quickly, safely and in a sustainable way. Continuous Delivery was born in the cloud and is a great way to get ideas to your customers. There’s one catch, if you want to adopt a Continuous Delivery strategy, you need to build applications differently, your team structure needs to change and how you test and validate systems needs to adapt to these changes.

Session #44 : Building 12 Factor Apps with Java by Kirk Knoernschild
The way we build and deliver software is changing. We must deliver software more quickly than ever before and traditional approaches to software architecture, infrastructure and methodology do not allow us to meet demand. We’ve reached the limits of agility through process improvement alone, and further increases demand we focus on improving architecture, infrastructure, and methodology simultaneously. 12 Factor is an app development methodology for building modern apps in the modern era.

Session #45 : Architecture as a business asset by Tudor Girba
Architecture is as important as functionality, at least in the long run. As functionality is recognized as a business asset, it follows that architecture is a business asset, too. In this talk we show how we can approach architecture as an investment rather than a cost, and detail the practical implications both on the technical and on the business level.

Session #46 : Architectural Trade-offs by Jeremy Deane
One cannot be all things to all people and the same is true of software architecture. There are inherent trade-offs that must be made of any architecture. Some architectural trade-offs are obvious such as performance versus security or availability versus consistency, while others are quite subtle such as resiliency versus affordability. This session will cover a number of architectural trade-offs and strategies for dealing with them.

Session #47 : Where Do Ideas Come From? Creating, Cultivating, and Communicating IP by Neal Ford
How do you create creativity? This talk offers techniques and perspectives to discover, grow, and project your ideas.
ArchConf
-Session Schedule-
(event schedule as of September 18, 2019)

Session #48 : How to Technology Good - Tips for Implementation at Scale by Laine Vyvyan and Josh Smith
Want to bring [new cool thing X] or [necessary technology change Y]? GOOD IDEA! Except...now what? If your company is more than about 3 people, how do you explain, enable, and encourage the adoption of this change, especially if it will require some work on everyone’s part?

6:30 - 7:30 PM : DINNER: OPAL BALLROOM

8:30 - 10:00 PM - Sessions

Session #49 : The Art of Problem Solving by Mark Richards
As Tech Leaders, we are presented with problems and work to find a way to solve them, usually through technology. In my opinion this is what makes this industry so much fun. Let's face it - we all love challenges. Sometimes, however, the problems we have to solve are hard - really hard. So how do you go about solving really hard problems? That's what this session is about - Heuristics, the art of problem solving. In this session you will learn how to approach problems and also learn techniques for solving them effectively. So put on your thinking cap and get ready to solve some easy, fun, and hard problems. Prerequisite: An open mind and a willingness to learn how to better approach and solve problems

Session #50 : Core Software Design Principles by Venkat Subramaniam
Creating code is easy, creating good code takes a lot of time, effort, discipline, and commitment. The code we create are truly the manifestations of our designs. Creating a lightweight design can help make the code more extensible and reusable.

Session #51 : Soft Skills for the Software Architect: Building Your Professional Network by Michael Carducci
You've heard the old adage "It's not what you know it's who you know." The focus of this session is divided between ways to better connect with everyone you meet as well as ways to grow your network, help and influence people and ultimately build long-term relationships and build your reputation.

Session #52 : Team Culture: Katakoda, a Learning Medium by Jonathan Johnson
We are continuously learning and keeping up with the changing landscapes and ecosystems in software engineering. Some technologies are difficult to learn or may take too much time for us to set up just to get to the key points of each technology. One of the reasons why you might be here at NFJS is to do exactly that -- too learn. Great! There are many mediums we use to learn and we often combine them for different perspectives. Books, how-to articles, GitHub readmes, blog entries, recorded talks on YouTube, and online courses. All these help us sort through the new concepts. I'm sure you have your favorites. Here we explore another learning medium to add to your toolbox: Katakoda

Session #53 : Extreme &#8216;Fake It Till You Make It&#8217; by Llewellyn Falco
Red - Green - (refactor) We all know that refactoring is suppose to be the step that let's us tease apart the logic of our code. ‘Fake it till you make it’ allows for us to evolve an emergent solution to complex problems. Yet this is usually glossed over when showing Test Driven Development.

Session #54 : Technology Innovation Diffusion by Jeremy Deane
In this session you will learn to strategically introduce technology innovations by applying specific change patterns to groups of individuals. Using these patterns and related techniques will not only benefit your organization but will ultimately benefit your career as a technologist by making you a better influencer, writer, and speaker.

Session #55 : Storytelling in a technical world by Tudor Gîrba
Our technical world is governed by facts. In this world Excel files and technical diagrams are everywhere, and too often this way of looking at the world makes us forget that the goal of our job is to produce value, not to fulfill specifications. Feedback is the central source of agile value. The most effective way to obtain feedback from stakeholders is a demo. Good demos engage. They materialize your ideas and put energies in motion. They spark the imagination and uncover hidden assumptions. They make feedback flow. But, if a demo is the means to value, shouldn't preparing the demo be a significant concern? Should it not be part of the definition of done?

Session #56 : Achieve Goals with SMART WINS by Christopher Judd
Everyone wants to be successful in life. Many have found the SMART (specific, measurable, achievable, relevant & time boxed) goal setting framework to be a powerful tool to help clarify and validate their goals. Unfortunately having well defined goals is not enough to obtain them. This is where WINS (write, incentivize, network & share) comes in.

Wednesday, Dec. 11

8:00 - 9:00 AM : BREAKFAST: OPAL BALLROOM
9:00 - 10:30 AM - Sessions

Session #57 : Choosing the Right Architecture Style by Mark Richards
Whether starting a new greenfield application or analyzing the vitality of an existing application, one of the decisions an architect must make is which architecture style to use (or to refactor to). Microservices? Service-Based? Microkernel? Pipeline? Layered? Space-Based? Event-Driven? SOA?. Having the right architecture style in place is essential to the success of any application, big or small. Come to this fast-paced session to learn how to analyze your requirements and domain to make the right choice about which architecture style is right for your situation.

Session #58 : Measuring Quality of design (1/2 day workshop) by Venkat Subramaniam
Before spending substantial effort in refactoring or altering design, it would be prudent to evaluate the current quality of design. This can help us decide if we should proceed with refactoring effort or a particular alteration of design. Furthermore, after evolving a design, using some design metrics would help us to evaluate if we have improved on the design front.

Session #59 : Kafka as a Platform: the Ecosystem from the Ground Up by Tim Berglund
Kafka has become a key data infrastructure technology, and we all have at least a vague sense that it is a messaging system, but what else is it? How can an overgrown message bus be getting this much buzz? Well, because Kafka is merely the center of a rich streaming data platform that invites detailed exploration.

Session #60 : Understanding Kubernetes: Fundamentals by Jonathan Johnson
So you have some code and it is in a bounded context with a REST API. You are on your way to Microservices. Next you wrap it in a container and now it is an image that others can run. Simple. Now what? No service is an island. Your service needs to log information, needs to scale and load balance between its clones. Your service needs environment and metadata way outside its context. What about where the service will run? Who starts it? What monitors its health? What about antifragility? Updates? Networking? Oh my. Services live in clusters and clusters live in data centers. Many concepts overlap with the features of cloud management. But don't get too flustered since, fundamentally, services are managed by clusters. There are several approaches to cluster management such as Docker Swarm, Mesos with Marathon and Kubernetes.

Session #61 : 12 (15) Factor App Workshop by Christopher Judd
Learn how to use Heroku's 12 (15) Factor App methodologies to make your applications more portable, scalable, reliable and deployable.

Session #62 : How to organize your code for long term success by Alexander von Zitzewitz
Most nontrivial software systems suffer from significant levels of technical and architectural debt. This leads to exponentially increasing cost of change, which is not sustainable for a longer period of time. The single best thing you can do to counter this problem is to give some love to your architecture by carefully managing and controlling the dependencies among the different elements and components of a software system. For that purpose we will introduce a DSL (domain specific language) that can be used to describe and enforce architectural blueprints. Moreover we will make an excursion into the topic of legacy software modernization.

Session #63 : Building Evolutionary Architectures Workshop by Neal Ford
This workshop highlights the ideas from the forthcoming Building Evolutionary Architectures, showing how to build architectures that evolve gracefully over time.

Session #64 : Finding Signal in the Noise: The art of Execution by Michael Carducci
In tech teams it's a constant firefight. We react. Then we react to the reaction... the cycle continues. In all this noise, in all this chaos, how do we move forward. How do we remain proactive?

10:30 - 11:00 AM : MORNING BREAK: OPAL BALLROOM FOYER

11:00 - 12:30 PM - Sessions

Session #65 : Applying Reactive Architecture Patterns by Mark Richards
Reactive architecture patterns allow you to build self-monitoring, self-scaling, self-growing, and self-healing systems that can react to both internal and external conditions without human intervention. These kind of systems are known as autonomic systems (our human body is one example). In this session I will show you some of the most common and most powerful reactive patterns you can use to automatically scale systems, grow systems, and self-repair systems, all using the basic language API and simple messaging. Through code samples in Java and actual run-time demonstrations, I'll show you how the patterns work and also show you sample implementations. Get ready for the future of software architecture - that you can start implementing on Monday.

Session #66 : Measuring Quality of design (1/2 day workshop) (continued) by Venkat Subramaniam
Before spending substantial effort in refactoring or altering design, it would be prudent to evaluate the current quality of design. This can help us decide if we should proceed with refactoring effort or a particular alteration of design. Furthermore, after evolving a design, using some design metrics would help us to evaluate if we have improved on the design front.
Session #67: Dissolving the Problem: Kafka is more ACID Than Your Database by Tim Berglund

It has become a truism in the past decade that building systems at scale, using non-relational databases, requires giving up on the transactional guarantees afforded by the relational databases of yore, ACID transactional semantics are fine, but we all know you can’t have them all in a distributed system. Or can we?

Session #68: Understanding Kubernetes: Container Patterns by Jonathan Johnson

Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Understanding Kubernetes: Fundamentals Aha moments with apps in containers can be quite liberating. The mobile space is saturated with “there’s an app for that”. For us, we now expect “there’s a container for that”. “Write once, run anywhere” (WORA) has changed to “Package once, run anywhere” (PORA). As the community of containers is riding up the hype curve we will look at some of those top aha moments together. • Go rouge with Java 9 and jlink • Polyglot microservices • RabbitMQ broker in 2 minutes • Private Docker hub in a container • Composing a Pod with multiple containers • Database flavors for integration testing The epiphanies come from the modular simplicity. Leveraging namespaces and using cgroups, these apps share a common kernel without polluting the host OS. This simplifies installation, conflicts and uninstalls. The barriers to getting something running are decreased and normalized to a container run command. This is subtly powerful and liberating. With this simplicity comes complexity such as shared resources, file systems, mounts, networking and overall cluster management.

Session #69: 12 (15) Factor App Workshop (continued) by Christopher Judd

Learn how to use Heroku’s 12 (15) Factor App methodologies to make your applications more portable, scalable, reliable and deployable.

Session #70: How to organize your code for long term success (continued) by Alexander von Zitzewitz

Most nontrivial software systems suffer from significant levels of technical and architectural debt. This leads to exponentially increasing cost of change, which is not sustainable for a longer period of time. The single best thing you can do to counter this problem is to give some love to your architecture by carefully managing and controlling the dependencies among the different elements and components of a software system. For that purpose we will introduce a DSL (domain specific language) that can be used to describe and enforce architectural blueprints. Moreover we will make an excursion into the topic of legacy software modernization.

Session #71: Building Evolutionary Architectures Workshop (continued) by Neal Ford

This workshop highlights the ideas from the forthcoming Building Evolutionary Architectures, showing how to build architectures that evolve gracefully over time.

Session #72: Leadership: How to be a Force Multiplier by Michael Carducci

Great leaders inspire, excite, and empower those in their teams. These leaders help create a team that is more than the sum of it’s parts; in short, a great leader can be a force multiplier for the team.

12:30 - 1:30 PM: LUNCH: OPAL BALLROOM

1:30 - 3:00 PM - Sessions

Session #73: Building Clustered Applications in a Distributed Architecture by David Sietz

For those who have built applications in a distributed architecture, we are all familiar with the difficulty of implementing clustering. Clustered application typically encounter the issues of synchronized communication and real-time maintenance (e.g.: configuration changes or updates) - both of which entail complex workarounds. But what if there were a simpler solution for building clustered applications - ones that don’t require a “management node” or downtime to make changes? How would it be designed and which patterns would be applied?

Session #74: Architectural Principles and Practices for building MicroServices by Venkat Subramaniam

Transitioning from a monolith to a microservices based architecture is a non-trivial endeavor. It is mired with many practices that may lead to a disastrous implementation if we’re not careful.

Session #75: Managing Schemas in Kafka by Tim Berglund

On the inside, Kafka is schemaless, but there is nothing schemaless about the worlds we live in. Our languages impose type systems, and the objects in our business domains have fixed sets of properties and semantics that must be obeyed. Pretending that we can operate without competent schema management does us no good at all.

Session #76: Kubernetes Operator Pattern by Jonathan Johnson

There are some straightforward and declarative techniques to request Kubernetes to run your applications. Typically, YAML declarations or Helm charts are sufficient to define your applications. Especially simple applications that are stateless. But more complex applications that are stateful, have persistence or change over time need a more dynamic management style. The Operator pattern has emerged as the way to control applications that not only run but also scale, backup, restore, adapt, update, and manage their more complex features. One fundamental nature of Kubernetes is its extensibility. By leveraging Custom Resource Definitions (CRDs) and extending the controllers, the Operator pattern was born.
ArchConf

_Session Schedule_
(event schedule as of September 18, 2019)

Session #77 : Architectural Anti-Pattern: Lock In & How to avoid it. by Llewellyn Falco
The single worst architectural anti-pattern is also the one I see the most often. It locks you into an architecture. Makes your choices permanent and inhibits being able to respond when you need to scale.

Session #78 : Software metrics for architects by Alexander von Zitzewitz
Software metrics can be used effectively to judge the maintainability and architectural quality of a code base. Even more importantly they can be used as “canaries in a coal mine” to warn early about dangerous accumulations of architectural and technical debt.

Session #79 : Modeling for Architects by Nathaniel Schutta
In some organizations, architects are dismissed as people that draw box and arrow diagrams - the dreaded whiteboard architect. While we don't want to foster that stereotype, it is important for an architect to be able to construct basic architectural diagrams. An architect must also be able to separate the wheat from the chaff eliminating those models that don't help tell the story while fully leveraging those that do.

Session #80 : Test Harnessing Legacy Infrastructure by Arty Starr
In the world of legacy code, we often end up inheriting a tangled ball of mess with a lack of automation, and no clear surfaces for testing. Yet still, under these circumstances, we're expected to safely make changes without regressions. Where do we start? How do we tackle this challenge? How do we get a handle on re-architecture? We'll start this discussion with a first-hand use case and example -- tackling the re-architecture of an 800k line JBoss application with near-zero unit tests. Ugh. The only option on the table was Selenium. UGH. Let's talk about alternative strategies. How have you tackled similar situations? How could we build a data-driven regression framework without going through the UI?

3:00 - 3:15 PM : BREAK: OPAL BALLROOM FOYER

3:15 - 4:45 PM - Sessions

Session #81 : Data as a Service Overview by David Sietz
[GitHub Repo](https://github.com/dsietz/daas) --- Should Information Management systems apply the services architecture? Many data provisioning and BI systems are monolithic, tightly coupled, difficult to scale, and stumble when it comes to delivering MVP in a timely manner.

Session #82 : Qualities of a Highly Effective Architect by Venkat Subramaniam
Many developers aspire to become architects. Some of us serve currently as architects while the rest of us may hope to become one some day. We all have worked with architects, some good, and some that could be better. What are the traits of a good architect? What are the skills and qualities we should pick to become a very good one? Come to this presentation to learn about things that can make that journey to be a successful architect a pleasant one.

Session #83 : Four Distributed Systems Architectural Patterns by Tim Berglund
Developers and architects are increasingly called upon to solve big problems, and we are able to draw on a world-class set of open source tools with which to solve them. Problems of scale are no longer consigned to the web's largest companies, but are increasingly a part of ordinary enterprise development. At the risk of only a little hyperbole, we are all distributed systems engineers now.

Session #84 : Understanding Kubernetes: Testing Patterns by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Understanding Kubernetes: Fundamentals. Highly cohesive and loosely coupled business functions can have a great impact on your agility to deliver new features. Microservices in containers is an effective implementation detail for continuous delivery. However, before you bite into that big sandwich, consider how provisioning a variety of data flavors as containerized endpoints could greatly improve your internal testing. How many times have you heard a colleague say, “Well that feature does not have integration tests because it requires a database with some specialized data”? Balderdash - put your data flavors in containers! Let's explore a solution to create a pipeline of data flavors. We use Docker images, Kubernetes Pods, Minikube to provision these endpoints. See how a Gradle project drives integration tests against these Pod endpoints, all ready for your continuous integration pipeline. In the end you can see the power of Consumer Driven Contracts against your dataset flavors.

Session #85 : Web Security Workshop by Brian Sletten
If you're not terrified, you're not paying attention. Publishing information on the Web does not require us to just give it away. We have a series of tools and techniques for managing identity, authentication, authorization and encryption so we only share content with those we trust. Before we tackle Web Security, however, we need to figure out what we mean by Security. We will pull from the worlds of Security Engineering and Software Security to lay the foundation for technical approaches to protecting our web resources. We will also discuss the assault on encryption, web security features and emerging technologies that will hopefully help strengthen our ability to protect what we hold dear.

Session #86 : Reflective Thinking by Tudor Girba
On the one hand, agile processes, like Scrum, promote a set of practices. On the other hand, they are based on a set of principles. While practices are important at present time, principles allow us to adapt to future situations. In this talk we look at Inspection and
Adaptation and construct an underlying theory to help organizations practice these activities. Why a theory? Because, as much as we want to, simply invoking "Inspect and Adapt" will not make it happen.

Session #87: Modeling for Architects (continued) by Nathaniel Schutta
In some organizations, architects are dismissed as people that draw box and arrow diagrams - the dreaded whiteboard architect. While we don't want to foster that stereotype, it is important for an architect to be able to construct basic architectural diagrams. An architect must also be able to separate the wheat from the chaff eliminating those models that don't help tell the story while fully leveraging those that do.

Session #88: Putting Out Fires with Gasoline by Arty Starr
Once upon a time, it was just me and my app -- the days when all I had to know was "get data, put on screen." Fast forward ten years later, and what the hell happened? The level of complexity that we deal with in modern software development is insane. Are we really better off than we were 10 years ago, or have we just been putting out our fires with gasoline?

4:45 - 5:00 PM: BREAK: OPAL BALLROOM FOYER

5:00 - 6:30 PM - Sessions

Session #89: Test Data Generation by David Sietz
[Github Repo](https://github.com/dsietz/test-data-generation) --- Continuous Integration has redefined our testing practices. Testing has become more focused, efficient, and re-positioned further upstream in the development lifecycle. Unfortunately, our testing systems haven't evolved in lock-step - specifically the provisioning of realistic test data. It remains common practice to extract, cleanse and load production data into our non-production environments. This is a lengthy process with serious security concerns, and still doesn't satisfy all our data content requirements. What if there is a better way of providing realistic test data? What if it could be generated on-demand as part of the Continuous Integration process - without the heavy databases and traditional batch jobs?

Session #90: Design Patterns in the Light of Lambda Expressions by Venkat Subramaniam
Design patterns are common place in OO programming. With the introduction of lambda expressions in languages like Java, one has to wonder about their influence on design patterns.

Session #91: Events, Dear Boy, Events by Tim Berglund
Harold McMillan was Prime Minister of England from 1957 to 1963, the last British PM born during Queen Victoria's rule, and one whose wit and even-keeled nature defined his administration. When asked by a reporter what might force his government off the course he had firmly laid out for it, he allegedly replied "Events, dear boy, events." The same might be said about what is driving software architectures today. Event-driven systems have enabled organizations to build substantial microservices ecosystems with all of the decoupling and evolvability that we were promised by the distributed computing technologies of 20 years ago. But these systems raise some interesting questions: if events now rule, what has become of entities? If we store events in logs, do we still need databases? Can we merely produce immutable events to trivially scalable logs and loose our microservices to consume them with no regard for what is actually out there in the world?

Session #92: Understanding Kubernetes: Serverless by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Understanding Kubernetes: Fundamentals From operating system on bare metal, to virtual machines on hypervisors, to containers orchestration platforms. How we run our code and bundle our applications continues to evolve. Serverless computing continuous our evolutionary path for our architectures.

Session #93: Web Security Workshop (continued) by Brian Sletten
If you're not terrified, you're not paying attention. Publishing information on the Web does not require us to just give it away. We have a series of tools and techniques for managing identity, authentication, authorization and encryption so we only share content with those we trust. Before we tackle Web Security, however, we need to figure out what we mean by Security. We will pull from the worlds of Security Engineering and Software Security to lay the foundation for technical approaches to protecting our web resources. We will also discuss the assault on encryption, web security features and emerging technologies that will hopefully help strengthen our ability to protect what we hold dear.

Session #94: McLuhan's Law by Tudor Gîrba
Marshall McLuhan told us among other things that "We shape our tools and thereafter our tools shape us." If this is true, we should be very careful with the tools that we expose ourselves to because they will determine the way we are going to think.

Session #95: An Architect's Guide to Site Reliability Engineering by Nathaniel Schutta
Development teams often focus on getting code to production losing site of what comes after the design and build phase. But we must consider the full life cycle of our systems from inception through deployment through to sunset, a discipline many companies refer to as site reliability engineering.
Session #96: DC/OS Deep Dive by Ken Sipe
In the container orchestration space, one of the top contenders is Apache Mesos and the Datacenter Operating System (DC/OS). This session will go into detail of each component of DC/OS along with how to use it. Anyone attending this session should be able to easily get started with DC/OS and have an understanding of what they would need to do to their application to enable it to be DC/OS friendly.

6:30 - 7:30 PM: DINNER: OPAL BALLROOM
7:30 - 9:00 PM: ARCHCONF 2019: EXPERT PANEL DISCUSSION - OPAL BALLROOM
9:00 - 10:30 PM: OUTDOOR RECEPTION: POOL LAWN

Thursday, Dec. 12
8:00 - 9:00 AM: BREAKFAST: OPAL BALLROOM
9:00 - 10:30 AM - Sessions

Session #97: Responsible Microservices (2019) by Nathaniel Schutta
These days, you can’t swing a dry erase marker without hitting someone talking about microservices. Developers are studying Eric Evan’s prescient book Domain Driven Design. Teams are refactoring monolithic apps, looking for bounded contexts and defining a ubiquitous language. And while there have been countless articles, videos, and talks to help you convert to microservices, few have spent any appreciable time asking if a given application should be a microservice. In this talk, I will show you a set of factors you can apply to help you decide if something deserves to be a microservice or not. We’ll also look at what we need to do to maintain a healthy micro(services)biome.

Session #98: Towards an Agile Design by Venkat Subramaniam
It's common knowledge: software must be extensible, easier to change, less expensive to maintain. But, how? That's what we often struggle with. Thankfully there are some really nice design principles and practices that can help us a great deal in this area.

Session #99: Software assessment: the discipline of technical decision making by Tudor Gîrba
Looking at what occupies most of our energy during software development, our domain is primarily a decision making business rather than construction one. As a consequence, we should invest in a systematic discipline to approach making decisions.

Session #100: Understanding Kubernetes: Continuous Pipelines by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Understanding Kubernetes: Fundamentals Kubernetes is a powerful platform for running containers and distributing computation workloads across resources. A significant question is how do you get all your code to this platform, continuously. In 2019 our community is bursting with new solutions to assist our delivery pipelines. While Jenkins is a dominant player, there is a growing array of new ideas and choices. From coding at your laptop to building containers to deployments, we will explore the various tools and techniques to reduce the delivery frictions.

Session #101: Machine Learning Workshop (continued) by Brian Sletten
Machine Learning is all the rage, but many developers have no idea what it is, what they can expect from it or how to start to get into this huge and rapidly-changing field. The ideas draw from the fields of Artificial Intelligence, Numerical Analysis, Statistics and more. These days, you'll generally have to be a CUDA-wielding Python developer to boot. This workshop will gently introduce you to the ideas and tools, show you several working examples and help you build a plan to for diving deeper into this exciting new field.

Session #102: Consumer Driven Contracts by Chris Maki
A real-world look at using Consumer Driven Contracts in practice. How to eliminate a test environment and how to build your services with CDC as a key component.

Session #103: Flying through Cloud Native (CNCF) by Ken Sipe
The maturing of industry projects and tools around cloud development and administration has led to the formation of the Cloud Native Computing Foundation. This new foundation is similar to the Apache Foundation in that it provides governance over projects from incubation to maturity. These projects define the current and future standards of the cloud which is important for all devops teams to be aware of. This session is a guided at jet speed tour of each project and how it fits in the eco-system.

Session #104: An Intro to Human Architecture: Populus ex Machina by Laine Vyvyan and Josh Smith
People are complicated. Understanding them is even MORE complicated. Understanding them when you’re inclined to introversion and IT is maybe the MOST complicated - which is a bummer, because we all have to somehow work together. However! Worry not.

10:30 - 10:45 AM: MORNING BREAK: OPAL BALLROOM FOYER
10:45 - 12:15 PM - Sessions

Session #105 : Responsible Microservices (2019) (continued) by Nathaniel Schutta
These days, you can’t swing a dry erase marker without hitting someone talking about microservices. Developers are studying Eric Evan’s prescient book Domain Driven Design. Teams are refactoring monolithic apps, looking for bounded contexts and defining a ubiquitous language. And while there have been countless articles, videos, and talks to help you convert to microservices, few have spent any appreciable time asking if a given application should be a microservice. In this talk, I will show you a set of factors you can apply to help you decide if something deserves to be a microservice or not. We’ll also look at what we need to do to maintain a healthy micro(services)biome.

Session #106 : Towards an Agile Design (continued) by Venkat Subramaniam
It’s common knowledge: software must be extensible, easier to change, less expensive to maintain. But, how? That’s what we often struggle with. Thankfully there are some really nice design principles and practices that can help us a great deal in this area.

Session #107 : Explainable software by Tudor Girba
Software systems should not remain black boxes. In this talk we show how we can complement domain-driven design with tools that match the ubiquitous language with visual representations of the system that are produced automatically. We experience of building concrete systems, and, by means of live demos, we exemplify how changing the approach and the nature of the tools allows non-technical people to understand the inner workings of a system.

Session #108 : Understanding Kubernetes: Meshing Around with Istio by Jonathan Johnson
Kubernetes out of the box is a strong platform for running and coordinating large collections of services, containers, and applications. As is, Kubernetes is powerful for many solutions. Remember Underdog? He was a mild-mannered dog, but when stress and conflict were introduced to the plot he took a magic pill, he became a superhero. Istio is a superhero for Kubernetes.

Session #109 : Machine Learning Workshop by Brian Sletten
Machine Learning is all the rage, but many developers have no idea what it is, what they can expect from it or how to start to get into this huge and rapidly-changing field. The ideas draw from the fields of Artificial Intelligence, Numerical Analysis, Statistics and more. These days, you’ll generally have to be a CUDA-wielding Python developer to boot. This workshop will gently introduce you to the ideas and tools, show you several working examples and help you build a plan to for diving deeper into this exciting new field.

Session #110 : Multi-Cloud Big Data by Dave Hendricksen
This session will focus on architecting for multi-cloud big data. This will include evaluating and comparing the big data capabilities in AWS and Azure, data synchronization, security, orchestration, disaster recovery, and other key aspects of multi-cloud enterprise big data systems.

Session #111 : The World is Blue/Green by Ken Sipe
One of the hardest activities and strategies of DevOps team or should we say production is how to transition from one version of an application to another version of an application with cascading consequences of service dependencies. There are a number of strategies for managing this concern. In this talk, we will outline a few of them along with required conditions of the underlying infrastructure to achieve it.

Session #112 : WYVIWYG - What you value is what you get by Chris Maki
You’ve heard of WYSIWYG, “what you see is what you get” but have you ever asked the question, “what do I value from my dev teams?” The cloud levels the playing field for many organizations, all companies using a public cloud platform have access to the same services and infrastructure as their competitors. The cloud makes it possible to trade capital expense for variable expense, increase your compute capacity by adjusting a configuration parameter, and provides you with a more secure and resilient compute platform than ever before. But everyone using a public cloud platform has access to these benefits. “What you value is what you get” looks at the unexpected results of putting emphasis on traditional software deliverables. In this presentation we will focus on undifferentiated work, how to recognize it and how to motivate your organization to focus instead on differentiated work. No one knows your customers better than you, why have your teams build out infrastructure, create software frameworks to manage dependency injection or write a framework to make building lambdas easier?

12:15 - 1:30 PM : LUNCH: POOL LAWN

1:30 - 3:00 PM - Sessions

Session #113 : Production Hardened Services by Nathaniel Schutta
By now I bet your company has hundreds, maybe thousands of services, heck you might even consider some of them micro is stature! And while many organizations have plowed headlong down this particular architectural path, your spidey sense might be tingling…how do we keep this ecosystem healthy?

Session #114 : Migrating to Java Modules: Why and How by Venkat Subramaniam
Java Modules are the future. However, our enterprise applications have legacy code, a lots of it. How in the world do we migrate from the old to the new? What are some of the challenges. In this presentation we will start with an introduction to modules and learn how to
create them. Then we will dive into the differences between unnamed modules, automatic modules, and explicit modules. After that we will discuss some key limitations of modules, things that may surprise your developers if they're not aware of. Finally we will discuss how to migrate current applications to use modules.

Session #115 : Steering Agile Architecture (continued) by Tudor Girba
“Emerge your architecture” goes the agile mantra. That’s great. Developers get empowered and fluffy papers make room for real code structure. But, how do you ensure the cohesiveness of the result? In this talk, we expose how architecture is an emergent property, how it is a commons, and we introduce an approach for how it can be steered.

Session #116 : Meshing Around with Observability by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes or Istio meshing be sure to attend: _Understanding Kubernetes: Fundamentals_ or _Understanding Kubernetes: Meshing Around with Istio_. Kubernetes is a complex container management system. Your application running in containers is also a complex system as it embraces the distributed architecture of highly modular and cohesive services. As these containers run, things may not always behave as smoothly as you hope. Embracing the notions of antifragility and designing a system to be resilient despite the realities of resource limitations, network failures, hardware failures and failed software logic. All of this demands a robust monitoring system to open views into the behaviors and health of your applications running in a cluster. Three important aspects to observe are log streams, tracing, and metrics.

Session #117 : Blockchain: Best or Worst Thing Ever? by Brian Sletten
If you listen to zealots and critics, blockchain-based systems and the cryptocurrencies they enable are either the Best Thing Ever or the Worst Thing Ever. As you may suspect, the reality is somewhere in-between. We will introduce the major ideas, technologies and players as well as evaluate them from technological, economic and social perspectives.

Session #118 : Deep Dive into AWS ML/AI Services by Dave Hendricksen
This session will be a deep dive into the machine learning and artificial intelligence services within AWS. This will include Amazon Comprehend, Forecast, Lex, Personalize, Polly, SageMaker, Recognition, Textract, Translate, and Transcribe. We will cover key concepts of each of the services, common use cases, and design patterns.

Session #119 : Microservices what a Servicemesh - The sequel. It is not just about cloud, it is about time by Emad Benjamin
Come to this session to learn about how we solved a fairly complex problem associated with maintaining predictable response time across set of service calls that are spread across multiple clouds. Many over the past few years have embraced microservices based architectures to increase flexibility and speed of feature delivery.

Session #120 : One of Us &#8211; the Importance of Community in Culture Change by Laine Vyvyan and Josh Smith
If companies truly want to go FAST, occasionally that requires changing something about the culture of the company. Processes get stale or overly complex, people don’t know why things are the way they are, and everyone wonders at the wisdom of asking too many questions.

3:00 - 3:15 PM : BREAK: OPAL BALLROOM FOYER

3:15 - 4:45 PM - Sessions

Session #121 : Thinking Architecturally by Nathaniel Schutta
Rich Hickey once said programmers know the benefits of everything and the trade offs of nothing...an approach that can lead a project down a path of frustrated developers and unhappy customers. As architects though, we must consider the trade offs of every new library, language, pattern or approach and quickly make decisions often with incomplete information. How should we think about the inevitable technology choices we have to make on a project? How do we balance competing agendas? How do we keep our team happy and excited without chasing every new thing that someone finds on the inner webs?

Session #122 : Towards an Evolutionary Architecture and Design by Venkat Subramaniam
Big up front design is discouraged in agile development. However, we know that architecture plays a significant part in software systems. Evolving architecture during the development of an application seems to be a risky business.

Session #123 : Steering Agile Architecture (continued) by Tudor Girba
"Emerge your architecture" goes the agile mantra. That’s great. Developers get empowered and fluffy papers make room for real code structure. But, how do you ensure the cohesiveness of the result? In this talk, we expose how architecture is an emergent property, how it is a commons, and we introduce an approach for how it can be steered.

Session #124 : Understanding Kubernetes: JVM Languages by Jonathan Johnson
Three evolutionary ecosystems work well together Java, Containers, and Kubernetes. A strong foundation of Java has always been write once, run anywhere, modularity, and polyglot. Containers and Kubernetes echo the exact same admirable qualities. Many of the patterns found in Java expand into Kubernetes as you massively scale your applications across the resources in your datacenter.
Containers entice us to pick up other languages and tools beyond the JVM. These new opportunities for additional languages and tools in containers need to be embraced. Does that mean you need to abandon Java? No, and with versions 9, 10, 11, 12… it continues to evolve, by design. The JVM continues as a first-class solution for your cloud-native adventures. Granted, Java <= 8 had a bumpy start with containers, but we have moved on from that. In this presentation, we see how to recent versions of Java run very effectively in containers, ideal for your Kubernetes based solutions.

Session #125 : The Decentralized Web by Brian Sletten
While the Web itself has strong decentralized aspects to how it is used, the backend technologies are largely centralized. The naming systems, the routing systems and the traffic that all points back to the same place for a website are all centralized technologies. This creates both a liability as well as a control point. In order to break free of some of these limitations, new technologies are emerging to provide a more decentralized approach to the Web.

Session #126 : Essential Skills for Software Architects by Dave Hendricksen
This session will focus on the essential skills that are needed by software architects on a daily basis from ideation to product delivery. For many architects, it's not the technology related areas that give you problems, but people related areas.

Session #127 : The Hybrid Cloud Runtime and the Rise of Application Platforms by Emad Benjamin
We often meet customers that have migrated to the public cloud only to later determine that some of their critical legacy application patterns have transitioned to a public cloud implementation, and they are now paying higher costs due to this design flaw. Regardless of cloud location, what really matters is how well you have abstracted the application platform nature of your enterprise workloads. If you don't understand your application workloads in terms of scalability, performance, reliability, security, and overall management, then you are simply shifting the problem from one cloud to another.

Session #128 : But I Don't WANNA - Protecting Your Resources as a Skilled Knowledge Worker by Laine Vyvyan and Josh Smith
We work in IT – and while we WORK with computers, we do not always FUNCTION like computers where inputs consistently make the same outputs. Our jobs are mostly theory and design and strategy, with some good old fashioned implementation thrown in – and as skilled knowledge workers, we function best when we respect that our mental and emotional resources matter.

4:45 - 5:00 PM : CONCLUSION OF ARCHCONF 2019 - THANK YOU FOR ATTENDING!