<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:00 - 9:00 AM</td>
<td>EARLY REGISTRATION: PRE-CONFERENCE WORKSHOP ATTENDEES ONLY - OPAL BALLROOM FOYER</td>
</tr>
<tr>
<td>9:00 - 6:00 PM</td>
<td>Architecture: The Hard Parts Neal Ford and Mark Richards</td>
</tr>
<tr>
<td></td>
<td>Applying Design Patterns Venkat Subramaniam</td>
</tr>
<tr>
<td></td>
<td>Developer To Architect Nathaniel Schutta</td>
</tr>
<tr>
<td></td>
<td>Kubernetes for Developers - 3 Day Deep Dive Jonathan Johnson</td>
</tr>
<tr>
<td></td>
<td>Agility Through Modular Architectures - From Modular Monoliths to Microservices Kirk Knorenschild</td>
</tr>
<tr>
<td></td>
<td>Hands-On Experience with Data as a Service David Sietz</td>
</tr>
<tr>
<td></td>
<td>Machine Learning Workshop Brian Sletten</td>
</tr>
<tr>
<td></td>
<td>Analyzing Architecture Alexander von Zitzewitz</td>
</tr>
<tr>
<td>5:00 - 6:30 PM</td>
<td>MAIN ARCHCONF REGISTRATION: OPAL BALLROOM FOYER</td>
</tr>
<tr>
<td>6:30 - 7:30 PM</td>
<td>OPENING NIGHT DINNER: OPAL BALLROOM</td>
</tr>
<tr>
<td>7:30 - 8:30 PM</td>
<td>Keynote: by Hans Dockter</td>
</tr>
<tr>
<td>8:30 - 10:30 PM</td>
<td>OPENING NIGHT OUTDOOR RECEPTION: POOL LAWN</td>
</tr>
<tr>
<td>Time</td>
<td>Session</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>7:30 - 8:00 AM</td>
<td>BREAKFAST &amp; LATE REGISTRATION: OPAL BALLROOM</td>
</tr>
<tr>
<td>8:00 - 10:00 AM</td>
<td>Microservices Migration Patterns Mark Richards</td>
</tr>
<tr>
<td>10:00 - 10:30 AM</td>
<td>MORNING BREAK: OPAL BALLROOM FOYER</td>
</tr>
<tr>
<td>10:30 - 12:00 PM</td>
<td>Microservices and Distributed Data Mark Richards</td>
</tr>
<tr>
<td>12:00 - 1:30 PM</td>
<td>LUNCH: OPAL BALLROOM</td>
</tr>
<tr>
<td>1:30 - 3:00 PM</td>
<td>Microservices Caching Strategies Mark Richards</td>
</tr>
<tr>
<td>3:00 - 3:15 PM</td>
<td>BREAK: OPAL BALLROOM FOYER</td>
</tr>
<tr>
<td>3:15 - 4:45 PM</td>
<td>The Rise and Fall of Microservices Mark Richards</td>
</tr>
<tr>
<td>4:45 - 5:00 PM</td>
<td>BREAK: OPAL BALLROOM FOYER</td>
</tr>
<tr>
<td>5:00 - 6:30 PM</td>
<td>Essential Architecture Practices Mark Richards</td>
</tr>
<tr>
<td>6:30 - 7:30 PM</td>
<td>DINNTER: OPAL BALLROOM</td>
</tr>
<tr>
<td>7:30 - 8:00 PM</td>
<td>Keynote: by Ken Spie</td>
</tr>
<tr>
<td>8:00 - 10:00 PM</td>
<td>The Art of Problem Solving Mark Richards</td>
</tr>
<tr>
<td>Time</td>
<td>Event</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>8:00 - 9:00 AM</td>
<td>BREAKFAST: OPAL BALLROOM</td>
</tr>
</tbody>
</table>
| 9:00 - 10:30 AM | Choosing the Right Architecture Workshops  
|                | Mark Richards                                                         
|                | Building Evolutionary Architectures Workshop                         
|                | Neal Ford                                                             
|                | Kafka as a Platform: the Ecosystem from the Ground Up                
|                | Tim Berglund                                                          
|                | Kubernetes Koncepts                                                  
|                | Jonathan Johnson                                                     
|                | Finding Signal in the Noise: The art of Execution                     
|                | Michael Carducci                                                     
|                | How to organize your code for long term success                      
|                | Alexander von Zitzewitz                                               
|                | Measuring Quality of design (1/2 day workshop)                       
|                | Verkat Subramaniam                                                   
|                | 12 (15) Factor App Workshop                                          
|                | Christopher Judd                                                     |
| 10:30 - 11:00 AM| MORNING BREAK: OPAL BALLROOM FOYER                                   |
| 11:00 - 12:30 PM| Applying Reactive Architecture Patterns Workshops                     
|                | Mark Richards                                                         
|                | Building Evolutionary Architectures Workshop (continued)             
|                | Neal Ford                                                             
|                | Dissolving the Problem: Kafka is more ACID Than Your Database         
|                | Tim Berglund                                                          
|                | Kubernetes Koncepts                                                  
|                | (continued)                                                          
|                | Jonathan Johnson                                                     
|                | Leadership: How to be a Force Multiplier (continued)                 
|                | Michael Carducci                                                     
|                | How to organize your code for long term success (continued)          
|                | Alexander von Zitzewitz                                               
|                | Measuring Quality of design (1/2 day workshop)                       
|                | (continued)                                                          
|                | Verkat Subramaniam                                                   
|                | 12 (15) Factor App Workshop (continued)                              
|                | Christopher Judd                                                     |
| 12:30 - 1:30 PM | LUNCH: OPAL BALLROOM                                                 |
| 1:30 - 3:00 PM  | Building Clustered Applications in a Distributed Architecture         
|                | David Sietz                                                           
|                | Modeling for Architects                                              
|                | Nathaniel Schutta                                                     
|                | The Database is Only Half Done: ksqlDB and Streaming Applications     
|                | Tim Berglund                                                          
|                | Architectural Principles and Practices for building MicroServices     
|                | Venkat Subramaniam                                                   
|                | Decomposing the Monolith                                             
|                | Chris Maki                                                           
|                | Software Metrics for Architects                                      
|                | Venkat Subramaniam                                                   
|                | Architectural Anti-Pattern: Lock In &amp; How to avoid it.            
|                | Llewellyn Falco                                                       |
| 3:00 - 3:15 PM  | BREAK: OPAL BALLROOM FOYER                                           |
| 3:15 - 4:45 PM  | Data as a Service Overview Workshops                                 
|                | David Sietz                                                           
|                | Modeling for Architects                                              
|                | (continued)                                                          
|                | Nathaniel Schutta                                                     
|                | Four Distributed Systems Architectural Patterns                      
|                | Tim Berglund                                                          
|                | Qualities of a Highly Effective Architect                            
|                | Venkat Subramaniam                                                   
|                | Micro-Service Orchestration Deep Dive                                 
|                | Ken Sipe                                                             
|                | Reflective Thinking                                                  
|                | Tudor Girba                                                          
|                | Konsumer Driven Kontracts                                           
|                | Jonathan Johnson                                                     
|                | Web Security Workshop                                                
|                | Brian Sletten                                                        |
| 4:45 - 5:00 PM  | BREAK: OPAL BALLROOM FOYER                                           |
| 5:00 - 6:30 PM  | Test Data Generation Workshops                                       
|                | David Sietz                                                           
|                | An Architect's Guide to Site Reliability Engineering                 
|                | Nathaniel Schutta                                                     
|                | Events, Dear Boy, Events Tim Berglund                                 
|                | Design Patterns in the Light of Lambda Expressions                   
|                | Venkat Subramaniam                                                   
|                | To 99.99 and Beyond                                                  
|                | Ken Sipe                                                             
|                | McLuhan&amp;rsquo;s Law                                               
|                | Tudor Girba                                                          
|                | Serverless Madness on Kubernetes                                     
|                | Jonathan Johnson                                                     
|                | Web Security Workshop                                                
|                | (continued)                                                          
|                | Brian Sletten                                                        |
| 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                                         |
## ArchConf
### Opal Sands Resort
### December 9 - 12, 2019

<table>
<thead>
<tr>
<th>Thu, Dec. 12, 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OPAL SUN</strong></td>
</tr>
<tr>
<td>8:00 - 9:00 AM</td>
</tr>
<tr>
<td>9:00 - 10:30 AM</td>
</tr>
<tr>
<td>10:30 - 10:45 AM</td>
</tr>
<tr>
<td>10:45 - 12:15 PM</td>
</tr>
<tr>
<td>12:15 - 1:30 PM</td>
</tr>
<tr>
<td>1:30 - 3:00 PM</td>
</tr>
<tr>
<td>3:00 - 3:15 PM</td>
</tr>
<tr>
<td>3:15 - 4:45 PM</td>
</tr>
<tr>
<td>4:45 - 5:00 PM</td>
</tr>
</tbody>
</table>
delivers faster and more scalable solutions to your customer.
on experience learning how the Rust language, a Kafka broker, and CouchDB cluster can be used to build out a DaaS system that
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-
data management, while avoiding many of the ant-patterns that traditional data provisioning and BI systems portray. Unlike traditional
Should Information Management systems apply the services architecture? Many data provisioning and BI systems are monolithic, tightly
Monoliths to Microservices by Kirk Knoernschild
Once you've broken apart your systems, how do you tie them back together? Putting them back together (Part 2) focuses on just this -
-Session Schedule-
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 @ OPAL SUN : Architecture: The Hard Parts by Neal Ford and Mark Richards
Software architects have no clean, easy decisions: everything is a terrible tradeoff. Architecture has lots of difficult problems, which
platform-agnostic class highlights by investigating what makes architecture so hard. This hands-on, platform-agnostic architecture
class goes beyond the fundamentals of software architecture and focuses on the really hard problems. This class focuses primarily on
aspects of distributed architectures and on areas surrounding modularity versus granularity, the challenges of event-driven architectures
(including difficult error handing issues), distributed architecture patterns and how to decouple services to achieve proper granularity.
Architecture is full of hard parts; by attending this workshop you can gain the insights and techniques to make it just a little softer. This
class is divided into 2 main parts: Part 1: Pulling Things Part (Modularity and Building Blocks) Part 2: Putting Them Back Together
(Patterns and Tradeoffs) Pulling things apart (Part 1) focuses on how to break up systems and identify levels of modularity. However,
once you've broken apart your systems, how do you tie them back together? Putting them back together (Part 2) focuses on just this -
the techniques of how to do orchestration, workflows, and transactions, while keeping parts decoupled from one another.

Session #2 @ OPAL SAND : 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 @ OPAL SEA : 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 work day 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 #4 @ BEACH VIEW/BREEZE : Kubernetes for Developers - 3 Day Deep Dive 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
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 #5 @ SEA SHORE/SEA WATCH : Agility Through Modular Architectures - From Modular Monoliths to Microservices by Kirk Knoernschild
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 #6 @ SAND DOLLAR : 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 of 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.
ArchConf
Session Schedule
(event schedule as of December 11, 2019)

Session #7 @ SAND DUNE: 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 #8 @ SUNSET: 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

Keynote: Developer Productivity at Scale: Using data and acceleration techniques to improve essential processes - Hans Dockter

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 - 10:00 AM: Sessions

Session #9 @ OPAL SUN: 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 @ OPAL SAND: 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 @ OPAL SEA: 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 #12 @ BEACH VIEW/BREEZE: 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 @ SEA SHORE/SEA WATCH: 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 #14 @ SAND DOLLAR: Adaptive Threat Modeling by Aaron Bedra
Security should always be built with an understanding of who might be attacking and how capable they are. Typical threat modeling exercises are done with a static group of threat actors applied in “best guess” scenarios. While this is helpful in the beginning, the real data eventually tells the accurate story. The truth is that your threat landscape is constantly shifting and your threat model should dynamically adapt to it. This adaptation allows teams to continuously examine controls and ensure they are adequate to counter the current threat actors. It helps create a quantitative driven approach to security and should be a part of every security teams tools.
ArchConf
-Session Schedule-
(event schedule as of December 11, 2019)

Session #15 @ SAND DUNE : 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 #16 @ SUNSET : Steering Agile Architecture by Tudor Gîrba
"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.

10:00 - 10:30 AM : MORNING BREAK: OPAL BALLROOM FOYER
10:30 - 12:00 PM - Sessions

Session #17 @ OPAL SUN : 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 @ OPAL SAND : 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 @ OPAL SEA : 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 #20 @ BEACH VIEW/BREEZE : 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 @ SEA SHORE/SEA WATCH : 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 #22 @ SAND DOLLAR : Intelligent Cache Systems by Aaron Bedra
Any system of significant scale or latency sensitivity employs the use of caching. It could be as simple as memoization, or as complicated as a fully distributed system. These ideas serve us well, but how do we take it to the next level? Join Aaron as he demonstrates customizing a caching system. He will discuss the pros and cons of embedding application and domain specificity into your caching model. Aaron will show a start to finish implementation of a custom Redis module that reduces latency, network round trips, and adds pub/sub notifications. Learn how to take your cache to the next level and encode elements of your system directly into the handling of your most accessed data.

Session #23 @ SAND DUNE : 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 #24 @ SUNSET : Beyond Technical Debt by Tudor Gîrba
"Technical debt" is a successful metaphor that exposes software engineers to economics, and managers to a significant technical problem. It provides a language that both engineers ("technical") and managers ("debt") understand. But, "technical debt" is just a metaphor that has its limitations, too. The most important limitation is that it presents a negative proposition: The best thing that can happen to you is having no technical debt.
12:00 - 1:30 PM : LUNCH: OPAL BALLROOM

1:30 - 3:00 PM - Sessions

**Session #25 @ OPAL SUN**: 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 @ OPAL SAND**: 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 @ OPAL SEA**: 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 #28 @ BEACH VIEW/BREEZE**: 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 @ SEA SHORE/SEA WATCH**: 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 #30 @ SAND DOLLAR**: Secrets Management by Aaron Bedra
We've all got secrets, but nobody seems to know where to put them. This long standing issue has plagued system design for ages and still has many broken implementations. While many consider this an application concern, the foundations rest in the design of the system. Join Aaron for an in-depth workshop that will cover the following secret management solutions: * Locally encrypted secrets with Ansible Vault * HSM backed local secrets with SOPS * AWS Secrets Manager * Hashicorp Vault Additionally, this workshop will demonstrate tools for discovering sensitive information checked in to your project.

**Session #31 @ SAND DUNE**: 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 #32 @ SUNSET**: 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?

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

3:15 - 4:45 PM - Sessions

**Session #33 @ OPAL SUN**: 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 @ OPAL SAND : 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 @ OPAL SEA : Automating Architecture Governance by Neal Ford
This session describes mechanisms to automate architectural governance at application, integration, and enterprise levels

Session #36 @ BEACH VIEW/BREEZE : Java MicroProfile - Enterprise Java Meets Microservices by Kirk Knoenrschild
Organizations have a lot of expertise in Java EE. With MicroProfile, developers can leverage this expertise to build cloud-native applications.

Session #37 @ SEA SHORE/SEA WATCH : 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 how to tackle the difficult areas. The end goal - enable anyone on your team to go from zero to productive in under 20 minutes.

Session #38 @ SAND DOLLAR : Secrets Management (continued) by Aaron Bedra
We've all got secrets, but nobody seems to know where to put them. This long standing issue has plagued system design for ages and still has many broken implementations. While many consider this an application concern, the foundations rest in the design of the system. Join Aaron for an in-depth workshop that will cover the following secret management solutions: * Locally encrypted secrets with Ansible Vault * HSM backed local secrets with SOPS * AWS Secrets Manager * Hashicorp Vault Additionally, this workshop will demonstrate tools for discovering sensitive information checked in to your project.

Session #39 @ SAND DUNE : 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 #40 @ SUNSET : The Business Agility Treasure Map: Charting a Course to Continuous Delivery by Laine Vyvyan and Josh Smith
All companies are IT companies. Except...not. All companies who feel the weight of their customers' ever-increasing demands for speed know that they SHOULD be IT companies, but most companies don't know how to get there - or even what "there" looks like.

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

5:00 - 6:30 PM - Sessions

Session #41 @ OPAL SUN : 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 @ OPAL SAND : 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 @ OPAL SEA : 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 December 11, 2019)

Session #44 @ BEACH VIEW/BREEZE : 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 @ SEA SHORE/SEA WATCH : 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 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 #46 @ SAND DOLLAR : Fluid Trust by Aaron Bedra  
Microservices bring about a series of architectural shifts. One of the most powerful is true separation of concerns. This change brings with it incredible security opportunities. Join Aaron as he demonstrates how to identify and execute on these opportunities. In this session you will explore service and data classification techniques, authentication and access control, and service interface design that respects classification boundaries. If you are interested in, building, or currently using Microservices, this session is a must see!

Session #47 @ SAND DUNE : 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 #48 @ SUNSET : To Hug or to Strangle: Tips for Dealing with Monolith Pain by Laine Vyvyan and Josh Smith  
A long time ago, in a land far far away, there were monoliths. These fabled artifacts brought consistency and stability to the land - but there was a cost in speed, agility, time, and development pain. Whether Java EE, .NET, or something else, the big ol' integrated plexi-purpose binaries or yore (and also now...) have grown into problems that hurt developers, architects, and the execution of business goals.

6:30 - 7:30 PM : DINNER: OPAL BALLROOM  
Keynote: Lessons Learned from the 737 Max - Ken Sipe

8:30 - 10:00 PM - Sessions  
Session #49 @ OPAL SUN : 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.

Session #50 @ OPAL SAND : 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 @ OPAL SEA : 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 @ BEACH VIEW/BREEZE : Writing Katacoda scenarios by Jonathan Johnson  
Explore another learning medium to add to your toolbox: **Katacoda**. This is a 90-minute mini-workshop where you learn to be an author on Katacoda. Bring your favorite laptop with just a browser and a text editor. Have a Github account and bring your laptop. Let's learn together.

Session #53 @ SEA SHORE/SEA WATCH : Lightning Talks by Llewellyn Falco  
Over the course of my life I have amassed a great quantity of 1-3 minute talks. Tonight we are going to Randomly pick from that list and see where the adventure takes us!
ArchConf
-Session Schedule-
(event schedule as of December 11, 2019)

Session #54 @ SAND DOLLAR : Machine Learning Strategies by Brian Sletten
Machine Learning is a key differentiator for modern organizations, but where does it fit into larger IT strategies? What does it do for you? How can it go wrong?

Session #55 @ SAND DUNE : 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 @ SUNSET : 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 @ OPAL SUN : Choosing the Right Architecture 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 @ OPAL SAND : 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 #59 @ OPAL SEA : 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 @ BEACH VIEW/BREEZE : Kubernetes Koncepts by Jonathan Johnson
You have some modular code with a REST API. You are on your way to Microservices. Next, you package it in a container image that others can run. Simple. Now what? 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. Don’t get flustered. We will explore how Kubernetes simplifies the complexity of distributed computing.

Session #61 @ SEA SHORE/SEA WATCH : 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?

Session #62 @ SAND DOLLAR : 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 @ SAND DUNE : 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 #64 @ SUNSET : 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.

10:30 - 11:00 AM : MORNING BREAK: OPAL BALLROOM FOYER
11:00 - 12:30 PM - Sessions

Session #65 @ OPAL SUN : 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 @ OPAL SAND : 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 #67 @ OPAL SEA : Dissolving the Problem: Kafka is more ACID Than Your Database by Tim Berglund
It has become at 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 @ BEACH VIEW/BREEZE : Kubernetes Koncepts (continued) by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Kubernetes Koncepts (1 of 2) 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). The growing community of containers is riding up the hype curve. We will look at many ways to assemble pods using architecture patterns you already know.

Session #69 @ SEA SHORE/SEA WATCH : 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.

Session #70 @ SAND DOLLAR : 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 costs 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 @ SAND DUNE : 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 #72 @ SUNSET : 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.
ArchConf
-Session Schedule-
(event schedule as of December 11, 2019)

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

1:30 - 3:00 PM - Sessions

Session #73 @ OPAL SUN : Building Clustered Applications in a Distributed Architecture by David Sietz
For those who have built applications in a distributed architecture, we are all to 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 version 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 @ OPAL SAND : 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 #75 @ OPAL SEA : The Database is Only Half Done: ksqlDB and Streaming Applications by Tim Berglund
In a world increasingly defined in software, is the database—a tool primarily built to aid human-computer interaction—always the right tool to choose? In this talk, we’ll look at a new type of database, built not only for the tables and columns we’re familiar with, but also the continuous, never-ending “streams of events” that represent data as it moves. We’ll take a look at ksqlDB’s syntax and show how it can replace bespoke Kafka Consumers with short, declarative queries.

Session #76 @ BEACH VIEW/BREEZE : 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 #77 @ SEA SHORE/SEA WATCH : Decomposing the Monolith by Chris Maki
There are distinct advantages to a monolithic architecture, but when does the balance tip towards smaller targets? What cultural and devops practices are essential to success? How do you decide where to make the first slice, evaluate and iterate? All these questions will be answered, and we’ll discuss pro tips from monolith-slaying case studies.

Session #78 @ SAND DOLLAR : 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 @ SAND DUNE : Kubernetes Operator Pattern by Jonathan Johnson
One fundamental attribute of Kubernetes is its extensibility. By leveraging Custom Resource Definitions (CRDs) and extending the controllers, the Operator pattern was born.

Session #80 @ SUNSET : 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.

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

3:15 - 4:45 PM - Sessions

Session #81 @ OPAL SUN : 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 @ OPAL SAND : 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 #83 @ OPAL SEA: 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 @ BEACH VIEW/BREEZE: 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 #85 @ SEA SHORE/SEA WATCH: Micro-Service Orchestration Deep Dive by Ken Sipe
Leading technical organizations in micro-service based architectures all use an orchestrator in their datacenter; be it Apache Mesos, Kubernetes, Tupperware, the Borg or Omega. The dominate platforms in the open source space are Kubernetes and Mesos. This session will dive deep into the core difference including: * Pod container lifecycle differences * In production scaling differences * Scheduling and orchestration management difference * Reconciliation management * Resource selection (affinity vs anti-affinity) * Service discovery This is NOT a one is better than the other smack-down. There are pro/con consequences for each orchestrator.

Session #86 @ SAND DOLLAR: 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 @ SAND DUNE: Konsumer Driven Kontrakts by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Kubernetes Kontrakts What kinds of frictions do you encounter when writing tests? Sometimes tests are never written because setting up a reliable and repeatable infrastructure is too difficult. With Kubernetes, you are now empowered to control your infrastructure that is reliable and repeatable. We will explore different testing techniques on Kubernetes, including an important one called “Consumer-Driven Contracts”.

Session #88 @ SUNSET: 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.

4:45 - 5:00 PM : BREAK: OPAL BALLROOM FOYER
5:00 - 6:30 PM - Sessions

Session #89 @ OPAL SUN: 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 life-cycle. Unfortunately, our testing systems haven’t evolved in lock-step - specifically the provisioning of realist 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 realist 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 @ OPAL SAND: 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 to deployment through to sunset, a discipline many companies refer to as site reliability engineering.

Session #91 @ OPAL SEA: 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 @ BEACH VIEW/BREEZE : 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 #93 @ SEA SHORE/SEA WATCH : To 99.99 and Beyond by Ken Sipe
When architecting a critical system the *A*vailability of CAP theorem becomes the most important element. Architecture measures availability in 9s with 99.99% equating less than 1 hour of unplanned downtime. This session will focus on what it takes to get there.

Session #94 @ SAND DOLLAR : McLuhan’s Law by Tudor Girba
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 @ SAND DUNE : Serverless Madness on Kubernetes by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Kubernetes Koncepts. 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 #96 @ SUNSET : 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.

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 @ OPAL SUN : Responsible Microservices 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 @ OPAL SAND : 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 @ OPAL SEA : Kubernetes Platform Security: A framework to think about it and some practical advice by Laine Vyvyyan and Josh Smith
Running containers requires a container management layer that thinks about and handles many moving parts. The architecture of that layer can’t have cliffs, gaps, or rough edges: it has to be a _reliable_ and _complete_ platform. One critical piece of this is security.

Session #100 @ BEACH VIEW/BREEZE : Software assessment: the discipline of technical decision making by Tudor Girba
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 #101 @ SEA SHORE/SEA WATCH: 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 #102 @ SAND DOLLAR: 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 #103 @ SAND DUNE: Kontinuous Pipelines on K8s by Jonathan Johnson
Prerequisite: If you are unfamiliar with Kubernetes be sure to attend: Kubernetes Koncepts. 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 #104 @ SUNSET: 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.

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

Session #105 @ OPAL SUN: Responsible Microservices (continued) by Nathaniel Schutta
These days, you can’t swing a dry erase marker without hitting someone talking about microservices. Developers are studying Eric Evans’ 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 @ OPAL SAND: 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 @ OPAL SEA: 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 #108 @ BEACH VIEW/BREEZE: Explainable software by Tudor Gîrba
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 experiences 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 #109 @ SEA SHORE/SEA WATCH: 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 @ SAND DOLLAR: 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 #111 @ SAND DUNE: 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. Istio is an open, platform-independent service mesh that manages communications between services in a transparent way. With a distributed architecture many things can fail, slow down and become less secure. Istio provides solutions to those stresses in our plot toward this architecture style:

- Traffic management
- Observability
- Policy enforcement
- Service identity and security

Session #112 @ SUNSET: WYVIWYG - What you value is what you get by Chris Maki
“What you value is what you get” looks at the unexpected results of emphasizing traditional software deliverables and organizational structures. 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 custom infrastructure or software frameworks instead of adding business value?

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

1:30 - 3:00 PM - Sessions

Session #113 @ OPAL SUN: Paved Roads - Architecting for Distributed Teams by Nathaniel Schutta
As we migrate towards distributed applications, it is more than just our architectures that are changing, so too are the structures of our teams. The Inverse Conway Maneuver tells us small, autonomous teams are needed to produce small, autonomous services. Architects are spread thin and can’t be involved with every decision. Today, we must empower our teams but we need to ensure our teams are making good choices. How do we do that? How do you put together a cohesive architecture around distributed teams?

Session #114 @ OPAL SAND: Migrating to Java Modules: Why and How by Venkat Subramaniam
Java Modules are the future. However, our enterprise applications have legacy code, a lot 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 @ OPAL SEA: 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 #116 @ BEACH VIEW/BREEZE: 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.

Session #117 @ SEA SHORE/SEA WATCH: Modern Software: WebAssembly in the Browser by Brian Sletten
Our industry never stops changing, but sometimes those changes are trivial and fluffly. Sometimes they are fundamental and enduring. This series is going to highlight some of the most important trends happening in the hardware, software, data and architecture spaces. While still new to most people, WebAssembly provides a formidable vision of safe, fast, portable code. Through clever choices and well-considered design, the basic vision allows us to target browsers as a platform using a variety of languages other than (but compatible with) Javascript. This technology coupled with advancements in the Web platform are setting up the future of Web-delivered applications to look more like (and likely to replace) desktop applications.

Session #118 @ SAND DOLLAR: 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 #119 @ SAND DUNE: 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
Session #120 @ SUNSET : Solving real problems without reading code by Tudor Gîrba
Too often, developers drill into the see of data related to a software system manually armed with only rudimentary techniques and tool support. This approach does not scale for understanding larger pieces and it should not perpetuate. Software is not text. Software is data. Once you see it like that, you will want tools to deal with it.

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

Session #121 @ OPAL SUN : Evolving to Cloud Native by Nathaniel Schutta
Every organization has at least a phalanx or two in the “Cloud” and it is, understandably changing the way we architect our systems. But your application portfolio is full of “heritage” systems that hail from the time before everything was as a service. Not all of those applications will make it to the valley beyond, how do you grapple with your legacy portfolio? This talk will explore the strategies, tools and techniques you can apply as you evolve towards a cloud native future.

Session #122 @ OPAL SAND : 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 @ OPAL SEA : 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 #124 @ BEACH VIEW/BREEZE : 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.

Session #125 @ SEA SHORE/SEA WATCH : 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 @ SAND DOLLAR : 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 #127 @ SAND DUNE : Distilling Java Containers by Jonathan Johnson
Three evolutionary ecosystems work well together Java, Containers, and Kubernetes. Past versions of Java were never designed to be “container aware”. This has led some to stray away from the JVM and consider other shiny languages. But wait, before you go, let’s discover what Java 9, 10, 11, 12, 13 (…) has done to get our applications into efficiently distilled containers that pack nicely into Kubernetes.

Session #128 @ SUNSET : Hands-on: Let’s save some Legacy Code by Llewellyn Falco
Like everyone else, you have a large product that is hard to work with. We’re going to change that in 75 minutes. Together we will save some gnarly legacy code (one thousand-line function). We will start with something hard to read, untested, and possibly buggy. We will finish with code that is stupidly easy to modify. You’ll learn 6 trivial techniques that you can apply over and over to fix 95% of the messiest code you have. You can take home this exercise to help the rest of your team learn these techniques. You’ll also learn how your team can teach itself a bunch more techniques to handle the other 5%. We are going to save some legacy code. In 90 minutes. While adding features. We will mob program; you will save this legacy code. We won’t introduce any bugs along the way. We will spend the time that you would normally use reading code to instead make it readable. You can apply these techniques and reduce the cost of coding within 48 hours of getting home. We have done this exercise with dozens of teams. They code differently now. Changing existing code is actually safer and cheaper than writing new code. Their designs get a little better each day. This session will improve your code and show you what skills to learn to gain further improvements.
ArchConf
-Session Schedule-
(event schedule as of December 11, 2019)

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