

salesforce

# Service Interoperability with gRPC

## gRPC in action

July 2020

**Varun Gupta, VP of Engineering**  
[linkedin.com/in/varungup](https://www.linkedin.com/in/varungup)

**Tuhin Kanti Sharma, Software Engineer**  
[linkedin.com/in/tuhin-kanti-sharma](https://www.linkedin.com/in/tuhin-kanti-sharma)



# Forward-Looking Statements



Statement under the Private Securities Litigation Reform Act of 1995:

This presentation contains forward-looking statements about the company's financial and operating results, which may include expected GAAP and non-GAAP financial and other operating and non-operating results, including revenue, net income, diluted earnings per share, operating cash flow growth, operating margin improvement, expected revenue growth, expected current remaining performance obligation growth, expected tax rates, the one-time accounting non-cash charge that was incurred in connection with the Salesforce.org combination; stock-based compensation expenses, amortization of purchased intangibles, shares outstanding, market growth and sustainability goals. The achievement or success of the matters covered by such forward-looking statements involves risks, uncertainties and assumptions. If any such risks or uncertainties materialize or if any of the assumptions prove incorrect, the company's results could differ materially from the results expressed or implied by the forward-looking statements we make.

The risks and uncertainties referred to above include -- but are not limited to -- risks associated with the effect of general economic and market conditions; the impact of geopolitical events; the impact of foreign currency exchange rate and interest rate fluctuations on our results; our business strategy and our plan to build our business, including our strategy to be the leading provider of enterprise cloud computing applications and platforms; the pace of change and innovation in enterprise cloud computing services; the seasonal nature of our sales cycles; the competitive nature of the market in which we participate; our international expansion strategy; the demands on our personnel and infrastructure resulting from significant growth in our customer base and operations, including as a result of acquisitions; our service performance and security, including the resources and costs required to avoid unanticipated downtime and prevent, detect and remediate potential security breaches; the expenses associated with new data centers and third-party infrastructure providers; additional data center capacity; real estate and office facilities space; our operating results and cash flows; new services and product features, including any efforts to expand our services beyond the CRM market; our strategy of acquiring or making investments in complementary businesses, joint ventures, services, technologies and intellectual property rights; the performance and fair value of our investments in complementary businesses through our strategic investment portfolio; our ability to realize the benefits from strategic partnerships, joint ventures and investments; the impact of future gains or losses from our strategic investment portfolio, including gains or losses from overall market conditions that may affect the publicly traded companies within the company's strategic investment portfolio; our ability to execute our business plans; our ability to successfully integrate acquired businesses and technologies, including delays related to the integration of Tableau due to regulatory review by the United Kingdom Competition and Markets Authority; our ability to continue to grow unearned revenue and remaining performance obligation; our ability to protect our intellectual property rights; our ability to develop our brands; our reliance on third-party hardware, software and platform providers; our dependency on the development and maintenance of the infrastructure of the Internet; the effect of evolving domestic and foreign government regulations, including those related to the provision of services on the Internet, those related to accessing the Internet, and those addressing data privacy, cross-border data transfers and import and export controls; the valuation of our deferred tax assets and the release of related valuation allowances; the potential availability of additional tax assets in the future; the impact of new accounting pronouncements and tax laws; uncertainties affecting our ability to estimate our tax rate; the impact of expensing stock options and other equity awards; the sufficiency of our capital resources; factors related to our outstanding debt, revolving credit facility, term loan and loan associated with 50 Fremont; compliance with our debt covenants and lease obligations; current and potential litigation involving us; and the impact of climate change.

Further information on these and other factors that could affect the company's financial results is included in the reports on Forms 10-K, 10-Q and 8-K and in other filings it makes with the Securities and Exchange Commission from time to time. These documents are available on the SEC Filings section of the Investor Information section of the company's website at [www.salesforce.com/investor](http://www.salesforce.com/investor).

Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements, except as required by law.



# THANK YOU



~\$ whoami



salesforce

- Always learning
- Fan of adventure sports
- Licensed pilot for 10+ years
- Riding (motorcycles) since before I ever drove a car
- Sailor in waiting



# Values Drive Value

Doing well and doing good

salesforce



**Leader in  
Innovation**

**FAST COMPANY**  
Best Workplaces  
for Innovators

**FORTUNE**  
Future 50 Top 10

**FORTUNE**  
100 Fastest Growing  
Companies

**LinkedIN**  
2019 Top  
Companies



**Leader in  
Philanthropy**

**People**  
#1 Company  
that Cares

**FORTUNE**  
Change the World

**SAN FRANCISCO  
BUSINESS TIMES**  
Top 75 Corporate  
Philanthropists



**Leader in  
Culture**

Great  
Place  
To  
Work.

World's Best  
Workplace

**FORTUNE**  
Best Companies  
to Work For

**indeed**  
Top Companies  
to Work For



# Innovating Infrastructure



We provide a simple and repeatable platform that delivers environments to Salesforce engineers for development, test, and production. The platform is extensible, scalable, and evolves to meet the needs of current and future instantiations across multiple substrates.

# Customer Revolution: We Are All Connected



Connected Apps & Products



Voice



Artificial Intelligence



Community



The Connected Customer



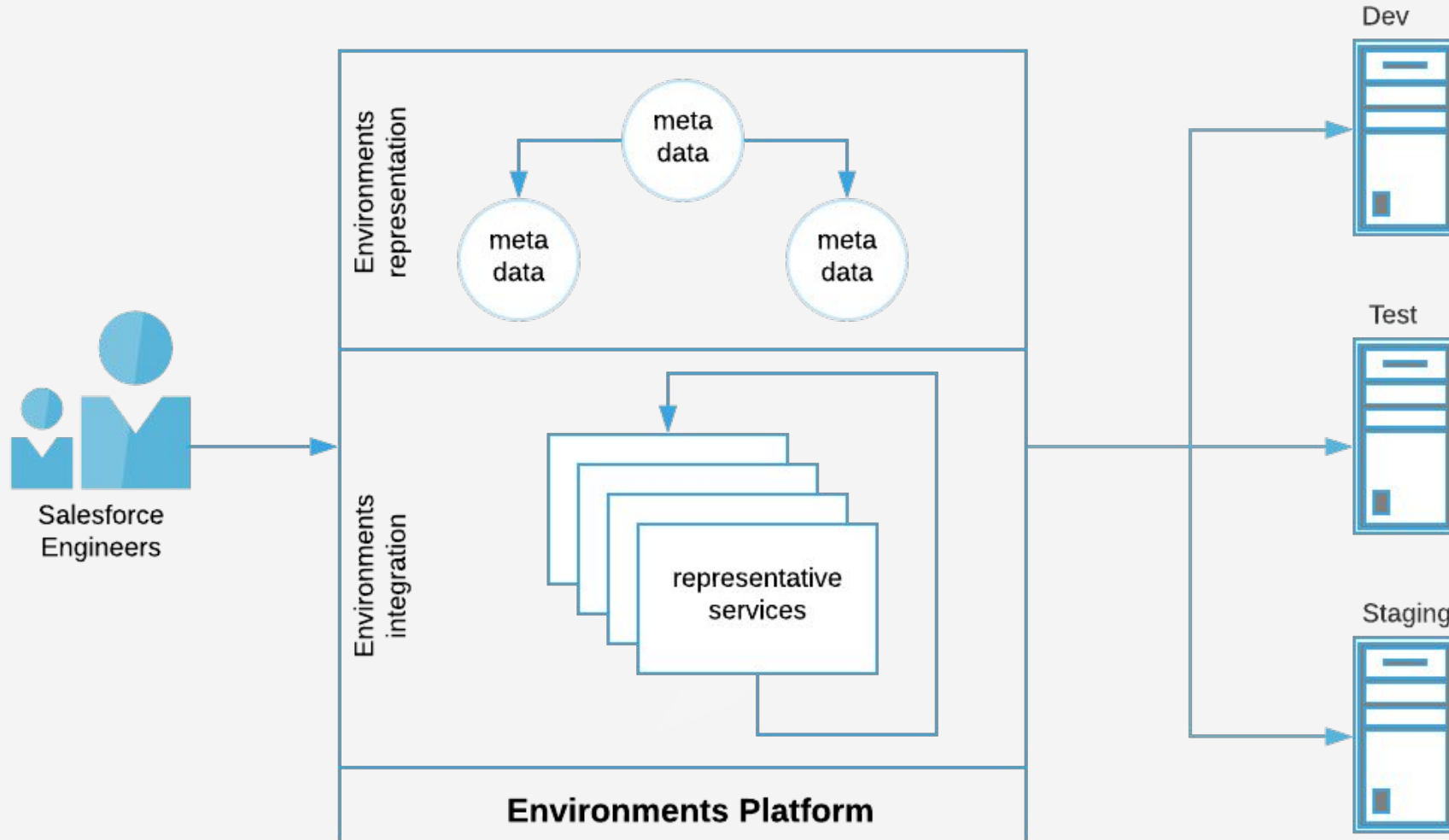
Blockchain



Trust



# Environments Platform

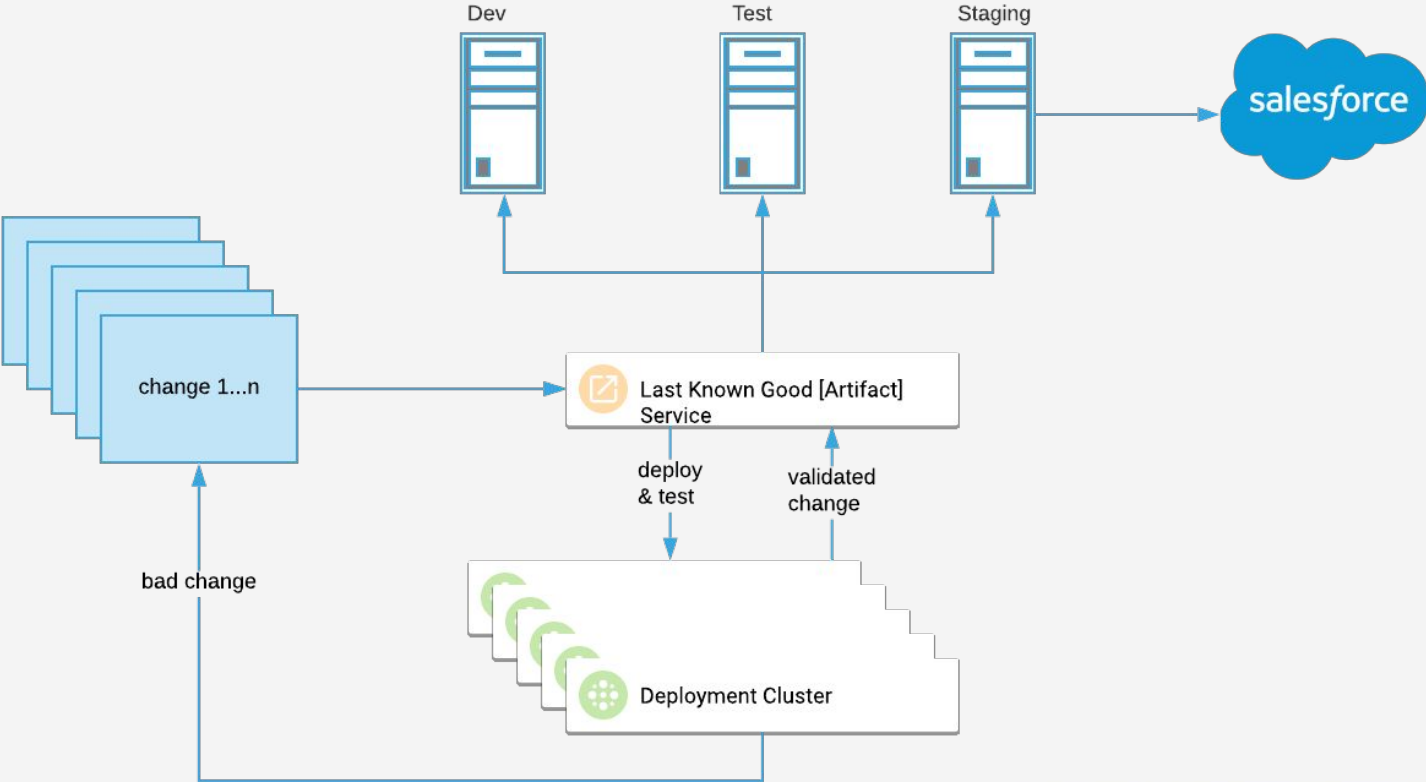




# Continuous Integration & Deployment

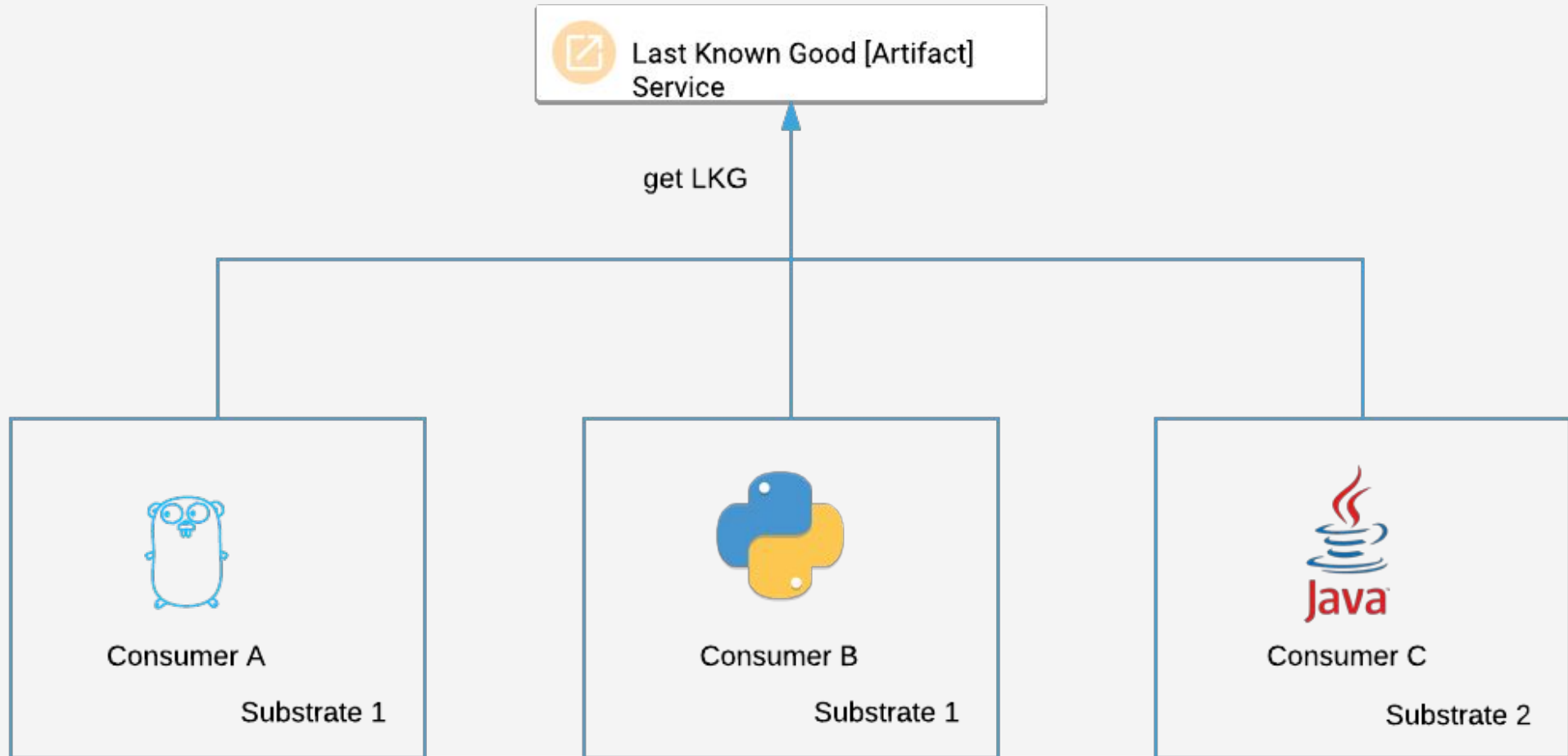
## Gatekeeping bad code for environments

Continuous deployment service that deploys and validates an artifact and its dependencies in a production like environment.



# Last Known Good [Artifact] Service

Requirement: Distributed polyglot clients



# Enabling Interoperability

## Achieving interoperability with gRPC

### Idiomatic Client / Server stubs

- protobuf: Common Interface Definition Language(IDL)

### Security & Authentication

- Native support for mTLS

### Resiliency & Scalability

- Consistent, reliable, low latency experience

### Open Source

- Collective innovation and support



**Tuhin Kanti Sharma**

Software Engineer

[linkedin.com/in/tuhin-kanti-sharma](https://www.linkedin.com/in/tuhin-kanti-sharma)



# ~\$ whoami



salesforce

- Outdoor enthusiast
  - Biked from Seattle to Portland
  - Hiked Mount Olympus in Greece
  - Learnt to ski watching YouTube videos
- Love to play sports (especially soccer)
- Enjoy volunteering and cooking
- Enjoy writing code and seeing it in action

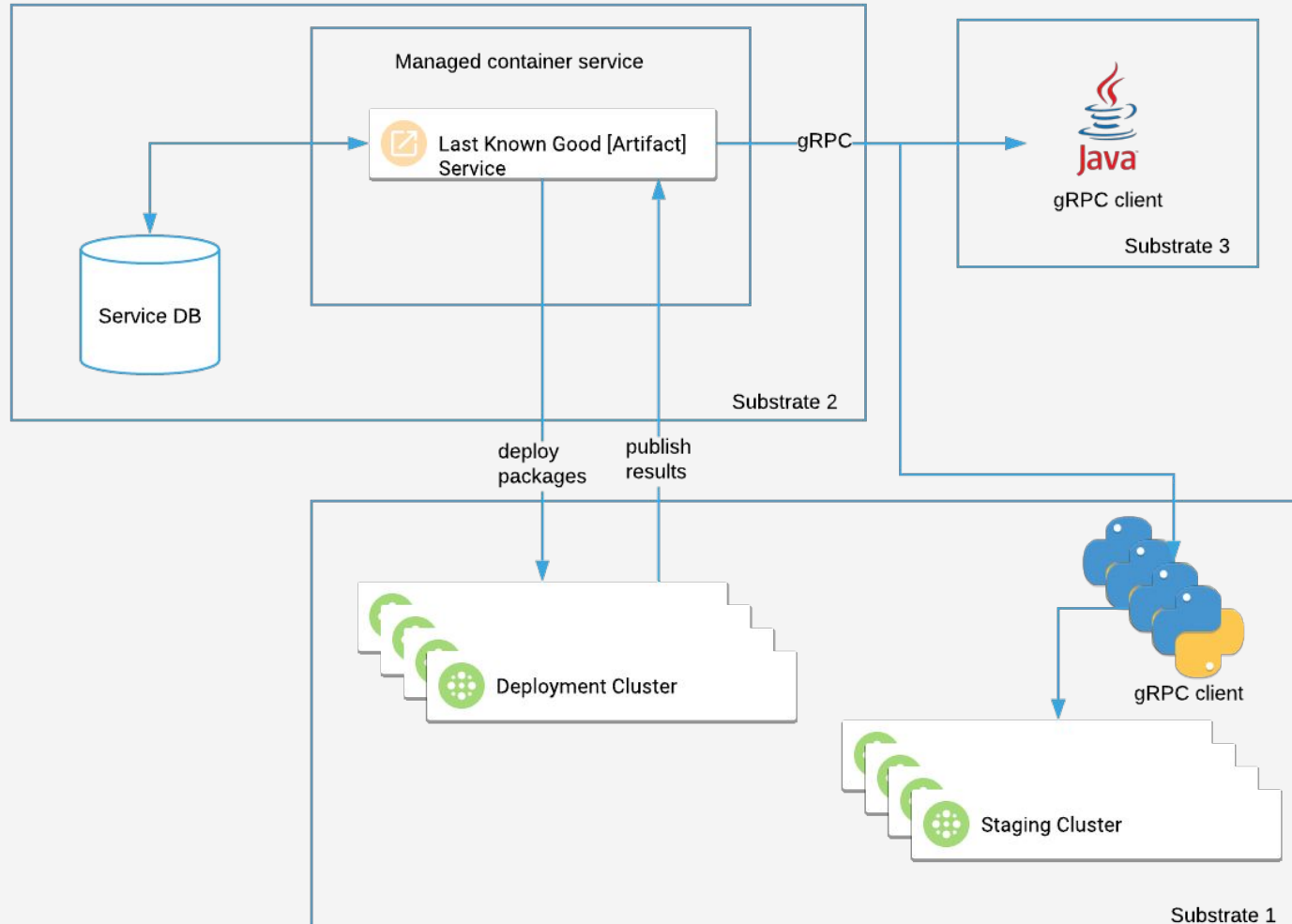


# Service Architecture



# Service Component Architecture

Interop with geo-distributed polyglot clients



# Interface Definition Language using protobuf

## Transparent API documentation and evolution

- Strongly typed native format
  - easily understood API spec
- Polyglot: generate stubs in any language
- Backwards/Forwards compatible schema evolution

```
message DeploymentConfig {
    string environment = 1;
    string branch = 2;
}

message Package {
    string manifest = 1;
    string product = 2;
    string branch = 3;
    string changelist = 4;
    string release = 5;
}

message PackageRequest {
    DeploymentConfig config = 1;
    oneof boundary {
        string from_manifest = 9;
        string from_changelist = 10;
        int32 limit = 11;
    }
}
```



# gRPC

fast, interoperable, bi-directional streaming, open source protocol



- Efficient (simple marshaling)
- Support for bidirectional streaming
- Simple Authentication API

```
service KnownGoodService {  
  
    rpc FindEnabledConfigurations  
      (google.protobuf.Empty) returns (stream  
      DeploymentConfig) {  
    }  
  
    rpc FindPassedPackages (PackageRequest)  
      returns (stream Package) {  
    }  
  
    rpc FindFailedPackages (PackageRequest)  
      returns (stream Package) {  
    }  
  
    rpc FindUnvalidatedPackages  
      (PackageRequest) returns (stream Package)  
    {  
    }  
}
```

The slide features a stylized illustration of a forest scene. Two large, dark brown trees with thick trunks and dense green foliage frame the top and sides of the image. In the background, there are rolling green hills and several tall, thin evergreen trees. The sky is a light, hazy blue. The overall style is clean and modern, typical of a corporate presentation.

# Salesforce Open Source Spring Boot Integration

# Service definition in Spring Boot

grpc-spring (com.salesforce.servicelibs)

```
import com.salesforce.grpc.contrib.spring.GrpcService;
```

```
@GrpcService
```

```
public class KnownGoodGrpcService extends KnownGoodService.KnownGoodServiceImplBase {
```

```
    private final DeploymentService deploymentService;
```

```
    private final PackageService packageService;
```

```
    public KnownGoodGrpcService(DeploymentService deploymentService,  
                                PackageService packageService) {
```

```
        this.deploymentService = deploymentService;
```

```
        this.packageService = packageService;
```

```
    }
```





# Stream Response

## Unary Streaming

```
/**
 * GRPC implementation for enabled deployment configs lookup
 * @param request          Empty request
 * @param responseObserver GRPC response observer
 */
@Override
public void findEnabledConfigurations (Empty request, StreamObserver<DeploymentConfig> responseObserver)
{
    try {
        Stopwatch stopwatch = Stopwatch.createStarted();
        Set<DeploymentConfig> enabledDeployments = deploymentService.findAllEnabledDeploymentConfigs();
        for (DeploymentConfig enabledDeployment : enabledDeployments)
            responseObserver.onNext(configToGrpcConfig(enabledDeployment));
        responseObserver.onCompleted();
    } catch (Exception e) {
        LOGGER.error(e.getStackTrace());
    }
}
```

# Reactive gRPC

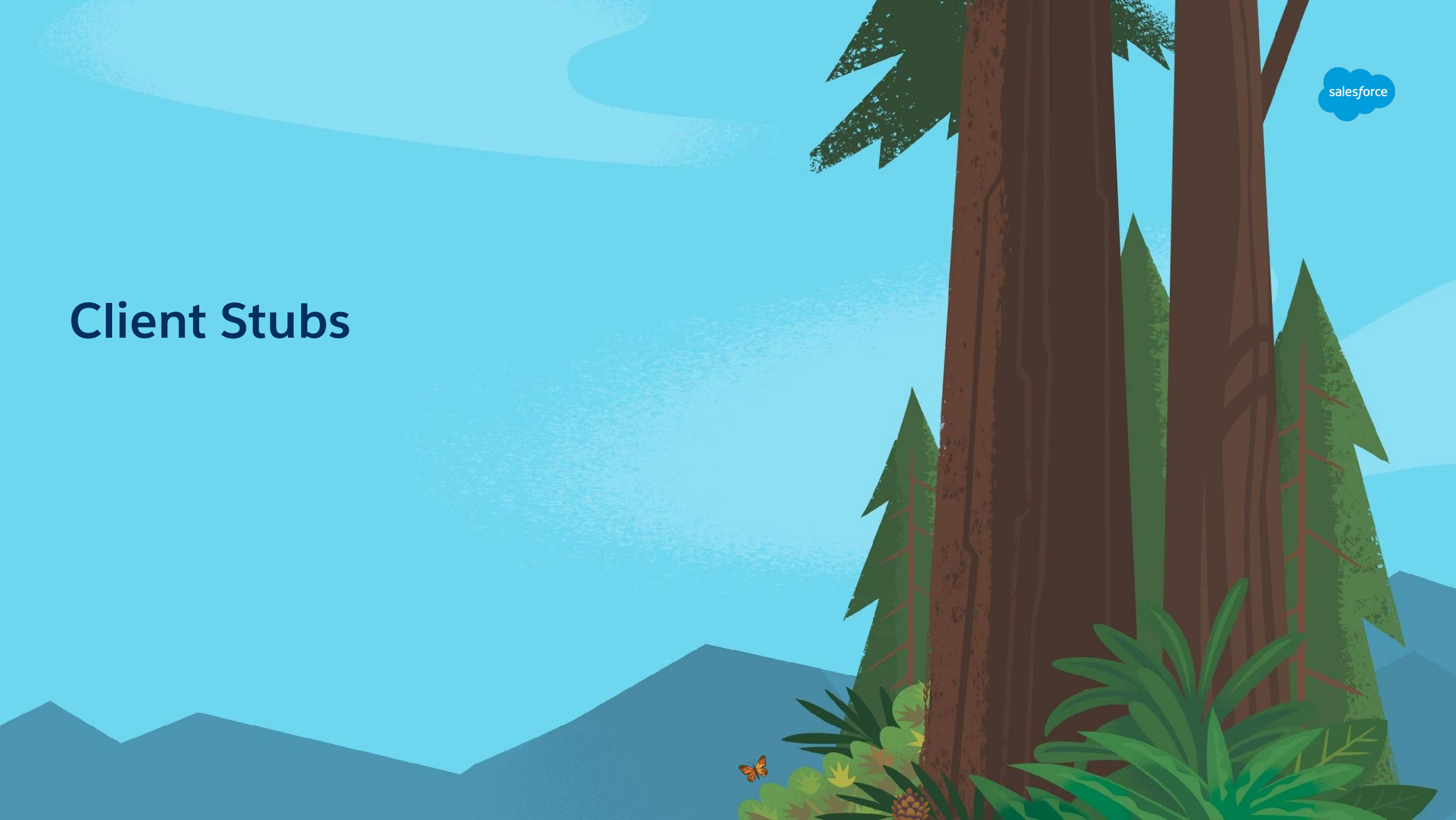
## Improve Availability with native flow control (reactive-grpc)

```
/**
 * GRPC reactive implementation for enabled deployment configs lookup
 */
@Override
public Flux<DeploymentConfig> findEnabledConfigurations(Empty request) {
    try {
        Stopwatch stopwatch = Stopwatch.createStarted();
        Set<DeploymentConfig> enabledDeployments = deploymentService.findAllEnabledDeploymentConfigs();

        metrics.enabledDeploymentsGrpcRequestTimeInMillis.updateValue(stopwatch.elapsed(TimeUnit.MILLISECONDS));
    } catch (Exception e) {
        handleException(e, responseObserver);
    }
    return Flux.fromIterable(enabledDeployments);
}
```



# Client Stubs





# Generate client stubs in Java and Python

Using protoc

Generate service definition for clients:

Java

```
protoc --plugin=protoc-gen-grpc-java \  
  --grpc-java_out="$OUTPUT_FILE" --proto_path="$DIR_OF_PROTO_FILE" "$PROTO_FILE"
```

Python

```
protoc --python-out="$SRC_PATH" $PROTO_FILE"
```



# Types of client stubs in Java

synchronous and asynchronous

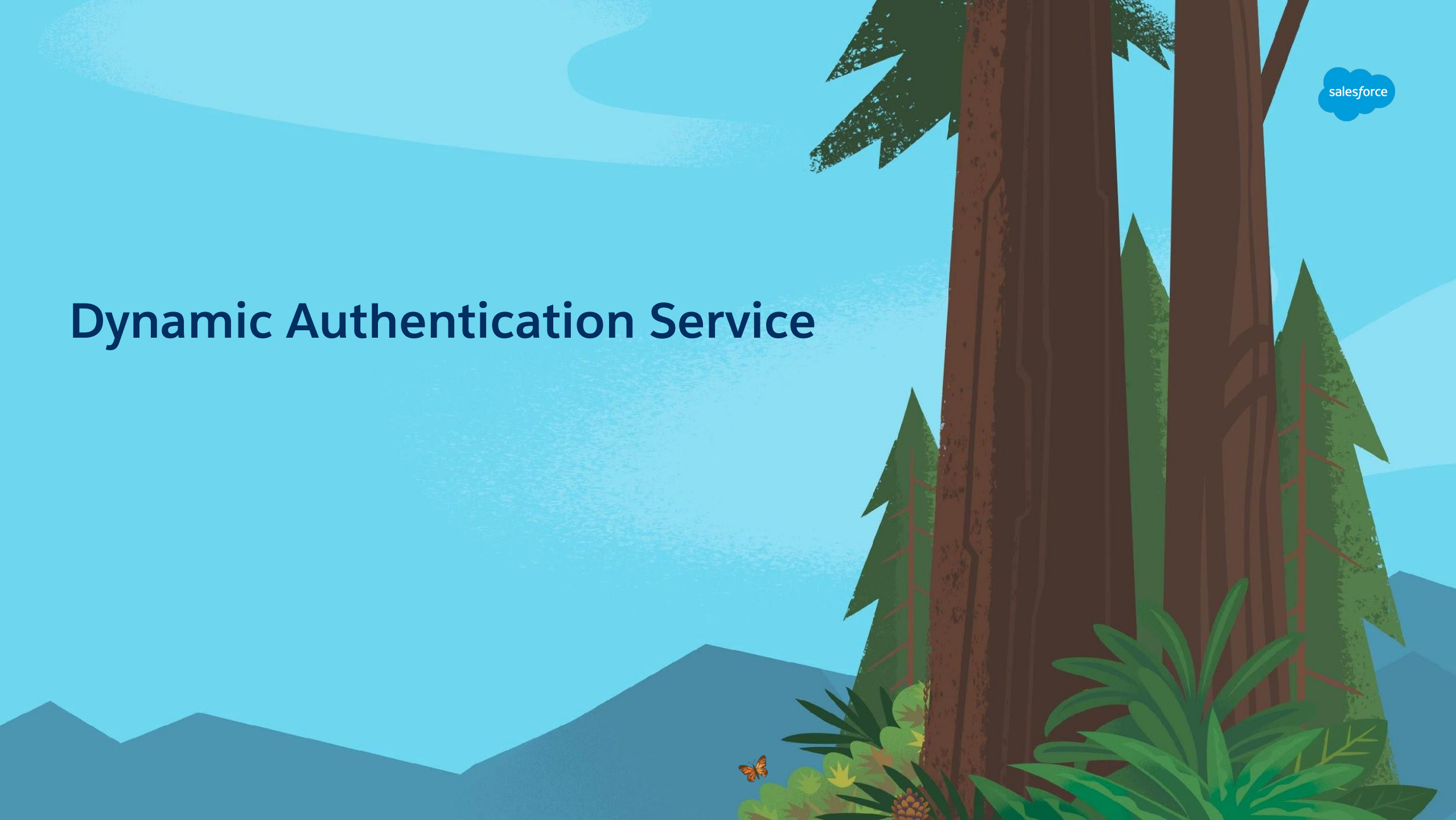
Auto-generated stubs are of 2 types:

```
//Synchronous(blocking) stub
KnownGoodServiceBlockingStub blockingStub = KnownGoodServiceGrpc.newBlockingStub(channel);
DeploymentConfig config = blockingStub.findEnabledConfigurations();
```

```
//Async stub
KnownGoodServiceStub asyncStub = KnownGoodServiceGrpc.newStub(channel);
StreamObserver<DeploymentConfig> responseObserver = new CapturingObserver();
asyncStub.findEnabledConfigurations(responseObserver);
```



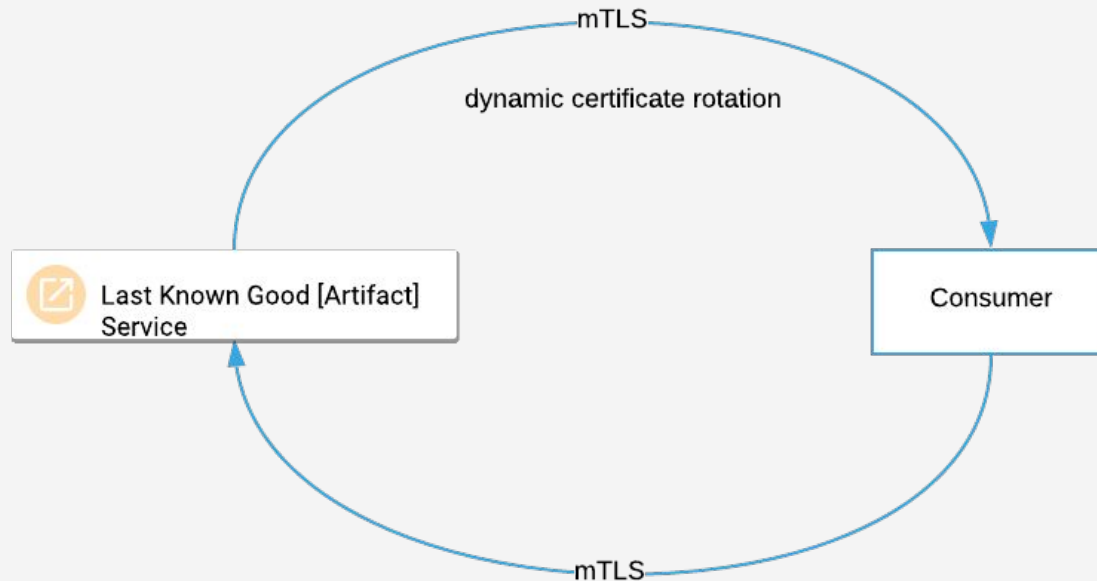
# Dynamic Authentication Service



# Service Authentication & Encryption

## mTLS with dynamic certificate rotation

- Transport level encryption
- Short lived certificates
- Automated provisioning and certificate rotation



```
apiVersion: v1
- name: grpc-spring-boot-service
count: 1
containers:
- name: grpc-service
image:
container-repo/grpc-service:1.0.3
# Mount volumes with certificates
volumeMounts:
- name: tls-client-cert
mountPath: /client-certs
- name: tls-server-cert
mountPath: /server-certs
# Add certificate volumes
volumes:
- name: tls-client-cert
pkiServiceCert:
type: client
- name: tls-server-cert
pkiServiceCert:
type: server
```

# Tying It Together

protobuf, gRPC, streaming, reactive, security

We demonstrated:

- protobuf
  - message format
  - service definition
- Salesforce open source Spring Boot libraries for
  - gRPC service
  - reactive gRPC service
- Client Stubs
  - protoc to generate in Java and Python
  - call using blocking and async
- Security using mTLS
  - dynamic cert rotation

# Key Takeaways

Learnings & Recommendations



# What did we learn?

## Triumphs & Challenges

### Triumphs

- Fairly smooth transition to gRPC
- Multisubstrate support
- Service reliability
  - Reduced Timeouts
  - Seamless transition to newer version

### Challenges

- Adoption is not automatic
- New technologies reference implementation



thank  
you

BLAZE  
YOUR  
TRAIL

salesforce

