



Swiss Re

Postgres-as-a-Service at Swiss Re

Dr. Andreas Geppert, Christoph Danielewicz, Franz Faul, Alfonso Tanga
Swiss Re Management
Andreas_Geppert@swissre.com

Contents

- ▶ Swiss Re Intro
- ▶ Database Strategy
- ▶ Postgres-as-a-Service and Database-as-a-Service at Swiss Re
- ▶ Postgres Applications
- ▶ Experiences and Open Issues
- ▶ Conclusion and Discussion

Swiss Re Group at a Glance

Swiss Re is a **leading and highly diversified global reinsurer**, founded in Zurich (Switzerland) in 1863

The Group offers **traditional reinsurance products and related services** for property and casualty, as well as for life and health businesses

The Group also offers **commercial insurance** through Corporate Solutions and manages **closed and open books of life and health business** via Life Capital

The financial strength¹ of the Swiss Re Group is currently rated: Standard & Poor's: AA- (stable); Moody's Aa3 (stable); A.M. Best: A+ (stable). Swiss Re Group's **Swiss Solvency Test Ratio** for 2017 is 262%

Swiss Re received a AAA sustainability rating from **MSCI** in May 2017

Key statistics (USD billions)	FY 2013	FY 2014	FY 2015	FY 2016
Gross premiums written	33.5	33.8	32.2	35.6
Net income	4.4	3.5	4.6	3.6
Shareholders' equity	33.0	35.9	33.5	35.6
ROE	13.7%	10.5%	13.7%	10.6%

¹ As at 31 December 2017

Swiss Re's Vision and Mission

OUR VISION

We make the world more resilient.

OUR MISSION

Together, we apply fresh perspectives, knowledge, and capital to anticipate and manage risk.

That's how we create smarter solutions for our clients, helping the world rebuild, renew, and move forward.

Swiss Re's Database Strategy

Key Messages (3 out of 6)

Deliver database-as-a-service

Provide open-source based alternative relational offerings

Broaden the scope to cover data management services beyond incumbent RDBMS

Database- and Postgres-as-a-Service

- ▶ Decision to built up a Postgres offering in Swiss Re (2017)
 - as a strategic, open source-based, cost-efficient option to closed source, commercial (relational) DBMS
 - NO (forced) exit out of existing commercial RDBMS
- ▶ Decision to build up Database-as-a-Service
 - based on (database) infrastructure by private cloud provider
- ▶ Build DBaaS integration and management on top of Postgres
 - e.g., internal DBaaS inventory and configuration management
- ▶ Pilot DBaaS with Postgres as first supported DBMS

Postgres-as-a-Service Enterprise Readiness Requirements

Functionality

Availability and DR

Security

Backup and restore

Scalability and performance

Monitoring

Operations and support

Release management

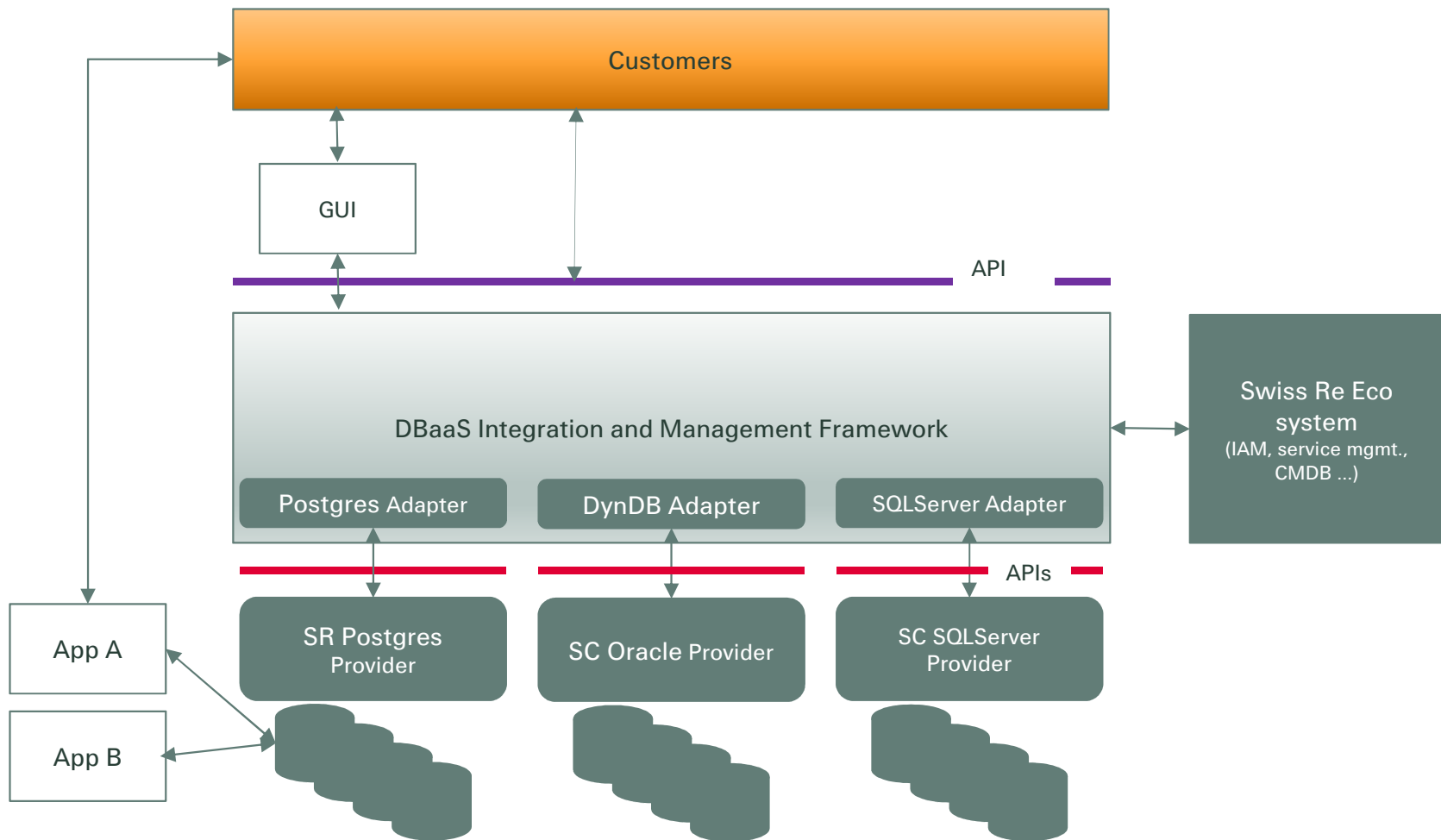
Migration support

Database-as-a-Service

High-level Service Catalogue

	Premium	Standard	Basic
Availability	Very high	High	normal
DR	RTO: fast RPO: zero/small	RTO: medium fast RPO: > 0h	RTO: b. e. RPO ≤ 1d
Maintenance	No/small downtime	Small downtime	Best effort
Scalability / Elasticity	Vertical and horizontal	Vertical	Vertical
Performance & Isolation	Isolation (compute, IO)	Shared, no isolation	Shared, no isolation
Support hours & reaction time	Comprehensive & fast	Normal	Best effort
Cost	High	Medium	Low

Swiss Re DBaaS Big Picture



DBaaS API: Database Creation

DELETE /roles/{guid} Drop a role

Additional Database Services

GET /databases Return databases

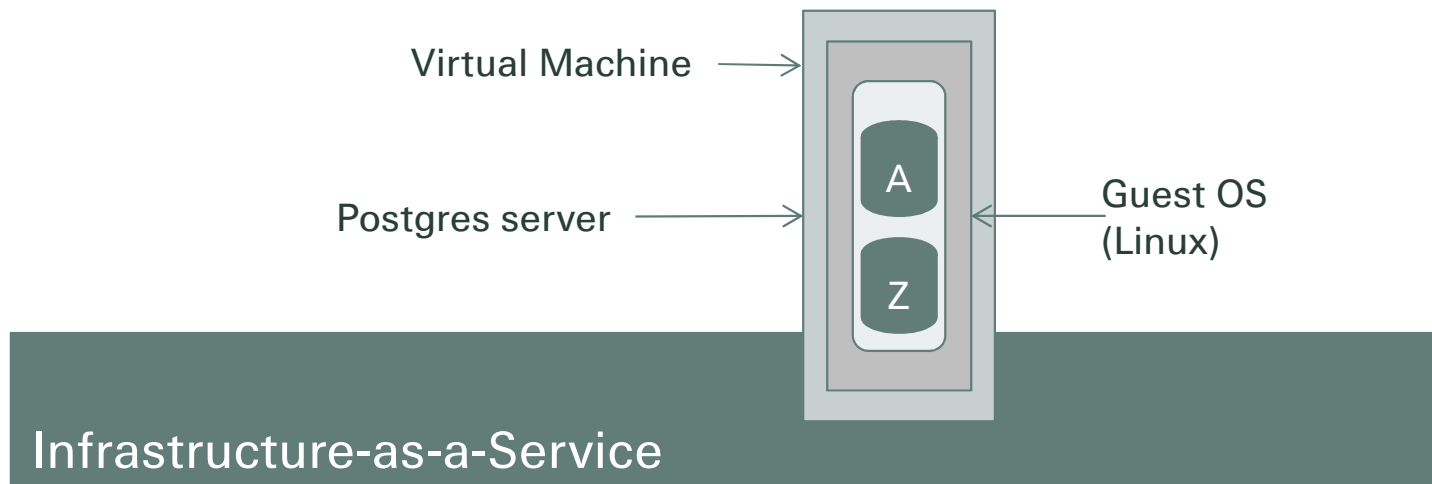
POST /databases Create a new database

Parameters Cancel

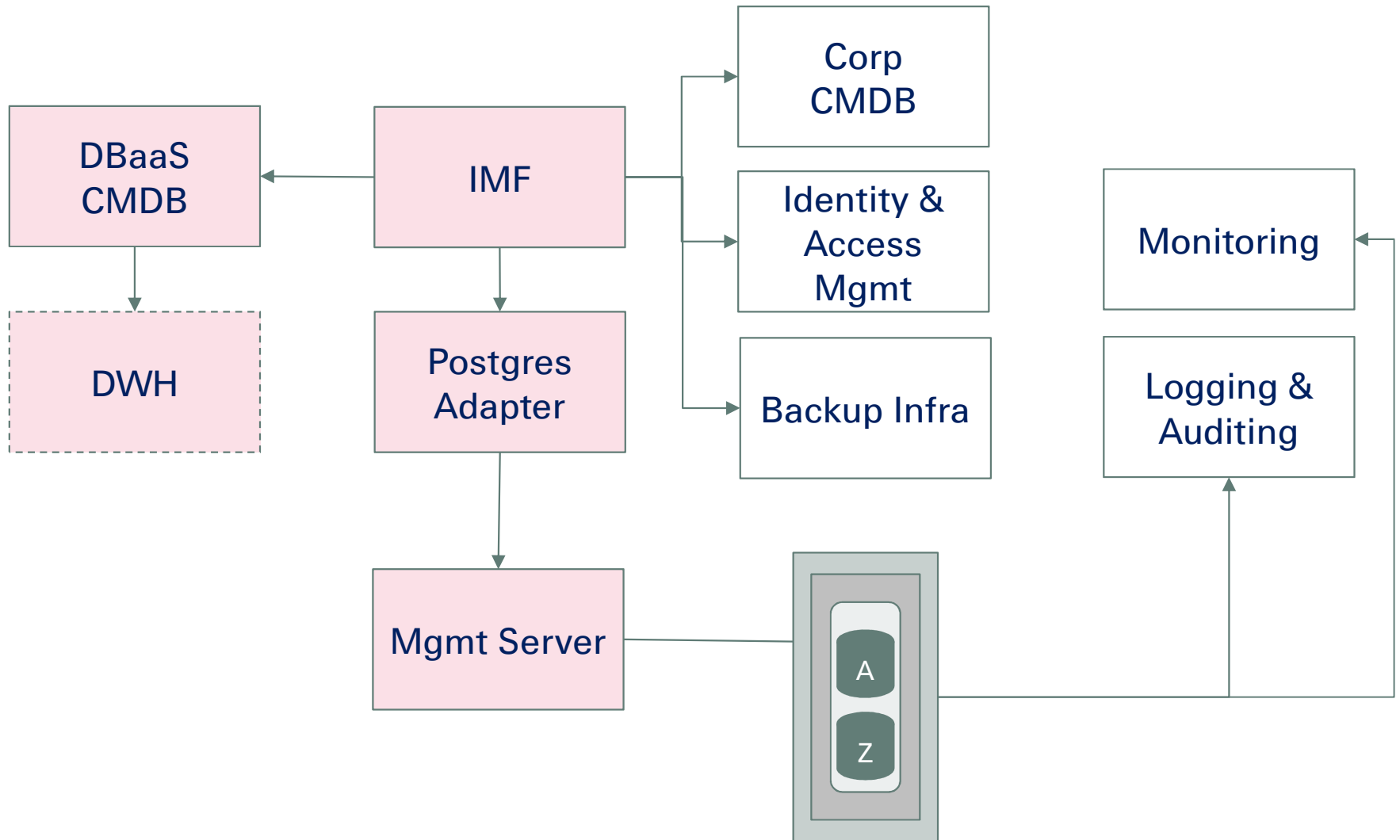
Name	Description
body <small>* required</small> (body)	The database definition Example Value Model <pre>{ "name": "PGDayDB", "environment": "dev", "dbms": "Postgres", "container": "qcp0000001", "appId": "PSQLC", "sensitiveData": "false", "collation": "C", "encoding": "UTF8", "description": "Database used in integration tests from bamboo", "maxsize": "50000", "owner": "s26xnc" }</pre>

Postgres-as-a-Service: Basic Setup

- ▶ Three different service classes are supported: basic, standard (DR), premium (HA & DR)
- ▶ DBaaS Postgres servers run in IaaS virtual machines (Linux)
- ▶ Each VM contains exactly one Postgres server
- ▶ A Postgres server hosts one or more application databases
- ▶ Postgres servers can be shared (i.e., contain databases from multiple applications) or be private (contain databases from a single application)



Postgres Server Creation Integration into Corporate Ecosystem



DBaaS Integration (1)

- ▶ Integration into corporate inventory and configuration management
- ▶ Integration into identity and access management
 - personal users: authentication via Active Directory
 - technical users: password authentication (SCRAM)
- ▶ Backup/restore/archive/clone integration
 - enterprise backup infrastructure preferred
 - BarMan under evaluation
- ▶ Patching/patch level monitoring and reporting
- ▶ Billing

DBaaS Integration (2)

▶ Monitoring integration

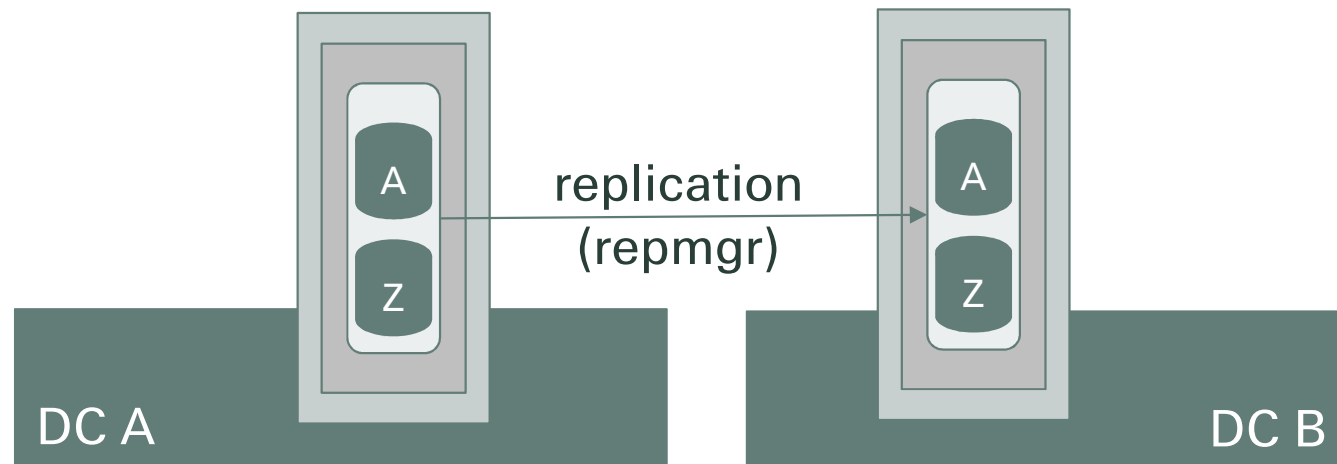
- DataDog (cloud-based SaaS solution)
- PGObserver and pg_analyze under evaluation

▶ Auditing and logging integration

- Logon/logoff events
- DDL events
- Server logs go into central ELK instance

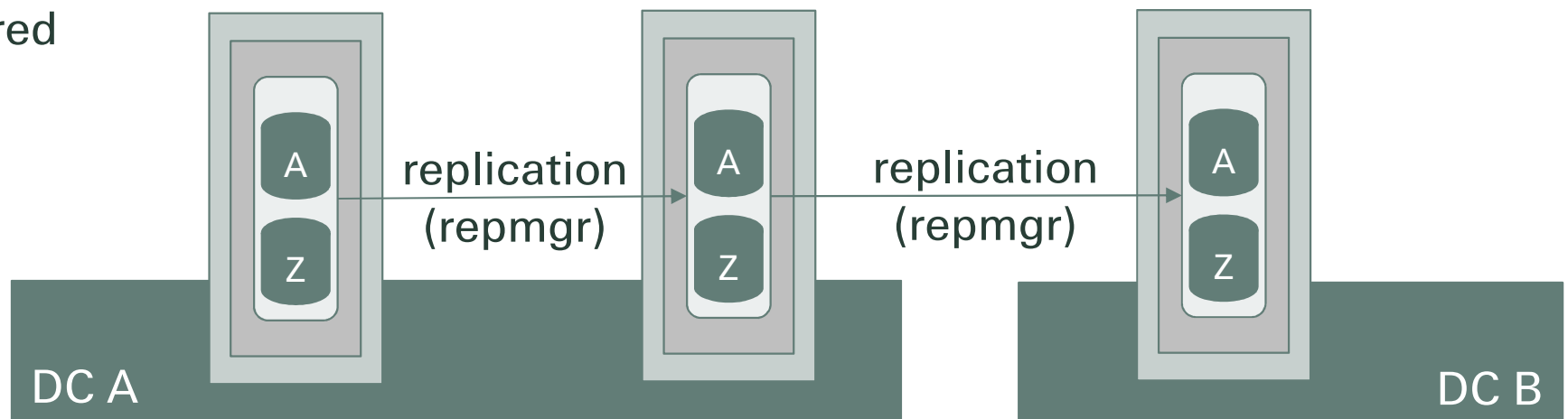
Postgres-as-a-Service: Standard Setup

- ▶ Disaster recovery provided through replication to a redundant Postgres server
- ▶ Standby server runs in a remote data center
- ▶ Replication is implemented with repmgr



Postgres-as-a-Service: Premium Setup

- ▶ The Premium service class adds another level of availability through cascading replication
- ▶ First standby runs in the same data center
 - failover in case of a local failure (VM, primary cluster)
- ▶ Second standby runs in the remote data center
 - failover in case of disaster
- ▶ Note: blueprint can be instantiated, but anti-affinities within DC cannot be ensured



Postgres Lifecycle Management Actions

- ▶ Create/read/update/delete for
 - servers
 - databases
 - schemas
 - users and roles as well as grants
- ▶ On-demand logical backup and restore
 - using `pg_dump`
- ▶ Export and import
- ▶ SQL execution
 - schema definition
 - direct data changes

Postgres Applications

DBaaS Inventory and CMDB

- generalized inventory and configuration of all DBaaS artefacts (clusters, databases, schemas, users, etc) and their relationships across all supported DBMSs

Postgres Infrastructure Inventory and CMDB (i.e., Postgres Adapter)

- detailed inventory and configuration of all Postgres artefacts (clusters, databases, schemas, users, etc) and their relationships

Oracle Cross-application dependency analysis

- Analysis of cross-schema and –application dependencies on database level for several hundred database applications

further internal (infrastructure) applications

business pilots

under construction: migration assessment tool

planned: capacity and performance management DWH

Experiences, Wishes, and Open Issues

- ▶ Postgres as a reliable and stable database platform
- ▶ Automated setup of blueprints meeting differentiated SLAs
- ▶ Usage of Postgres for both, as provider and platform (for CMDB etc) enabled us to start much earlier and progress further than would have been possible otherwise
- ▶ Wishes for upcoming releases
 - Security: authorization against AD
 - Metering – to break down server cost (compute resources) to individual databases)

An aerial photograph of a tennis court. A curved concrete path runs along the left and top edges of the court. A black chain-link fence follows the right and bottom edges. The court is surrounded by green grass and trees. The text "Questions? Comments?" is overlaid on the left side of the image.

Questions?
Comments?



Legal notice

©2018 Swiss Re. All rights reserved. You are not permitted to create any modifications or derivative works of this presentation or to use it for commercial or other public purposes without the prior written permission of Swiss Re.

The information and opinions contained in the presentation are provided as at the date of the presentation and are subject to change without notice. Although the information used was taken from reliable sources, Swiss Re does not accept any responsibility for the accuracy or comprehensiveness of the details given. All liability for the accuracy and completeness thereof or for any damage or loss resulting from the use of the information contained in this presentation is expressly excluded. Under no circumstances shall Swiss Re or its Group companies be liable for any financial or consequential loss relating to this presentation.