AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

FSI308

Globe Life: Accelerate innovation with self-service analytics

Srikanth Sopirala
Principal Analytics Specialist Solutions Architect
AWS

Arthur Heezen
Director of Data Strategy and Services
Globe Life

Dileep Kumar
Senior Manager, Data Engineering and Cloud Data Platforms
Globe Life

Agenda

Self-service analytics on AWS

Globe Life story

- Objectives/challenges
- AWS solution
- Achievements/lessons learned

Q&A/wrap up

Self-service analytics on AWS

What is self-service analytics?

A. Like an IT vending machine

B. “Get IT out of my way!”

C. Running reports on my own

D. Empowers business users to concentrate on core analytics tasks on their own, including data exploration, verification, and visualization

Accelerate innovation

Reduce IT dependence

Upskill the business

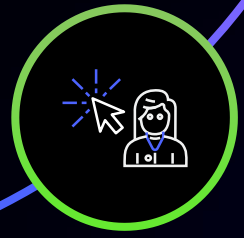
Self-service applies across multiple personas

Data engineers/database developers

Data analysts

Data scientists

BI professionals



Administrators

aws

Self-service analytics is challenging

Multiple analytics needs

Increasing and unpredictable costs

Difficult to manage systems

Self-service analytics

Inflexible tools and environments

Security and compliance risks

Self-service analytics requirements

Democratize access

Data

Tools

Training

Maintain control

Security

Governance

Compliance

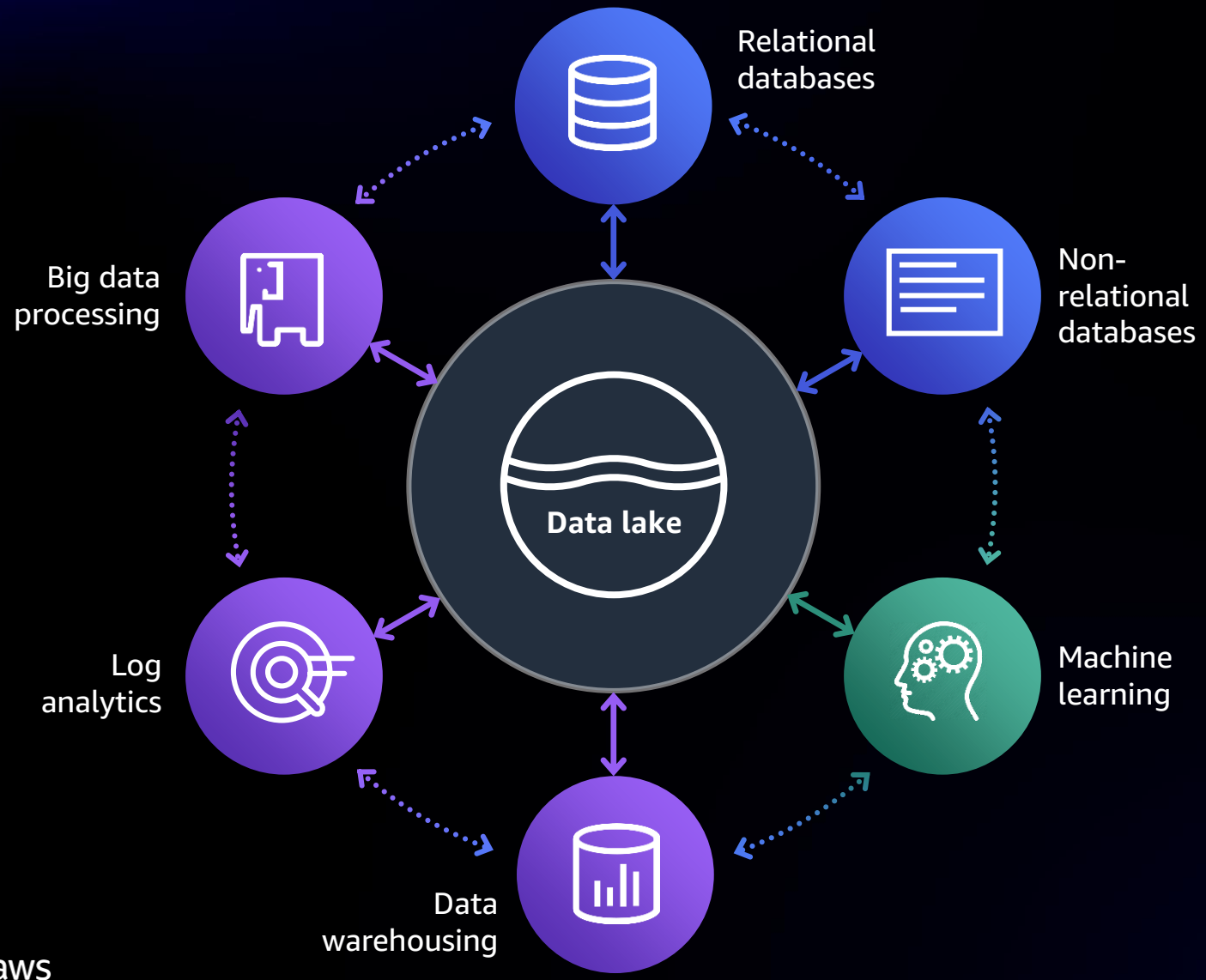
Drive orders of magnitude change

Measurable success

Performance/cost

Administrative ease

Modern data architecture: AWS point of view



Scalable data lakes

Purpose-built data services

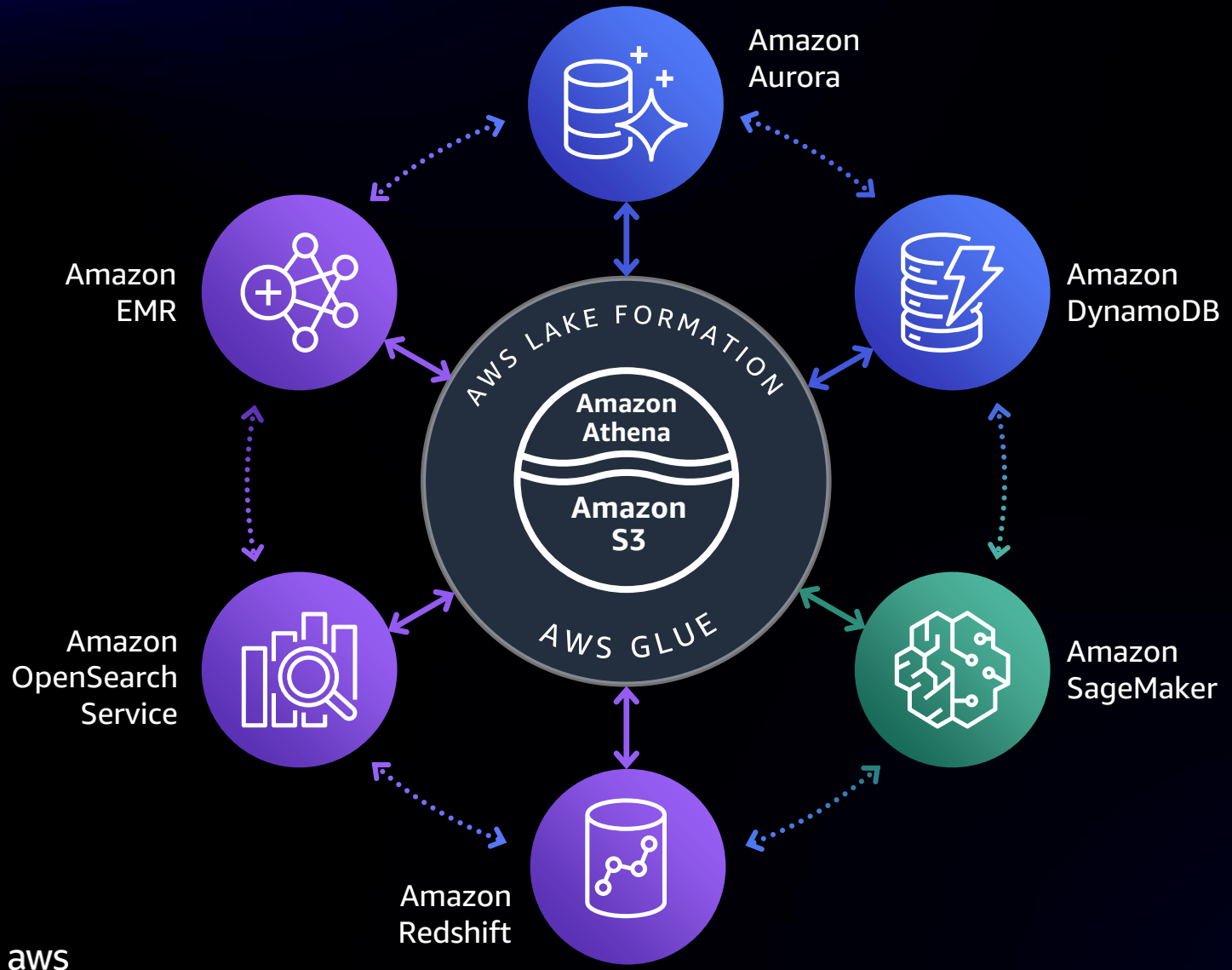
Seamless data movement

Unified governance

Performant and cost effective

aws

Self-service with AWS



Simplify administration

Visual tools accelerate adoption and productivity

Data and tool integration

Centralized security and governance

Globe Life's self-service analytics journey

About Globe Life

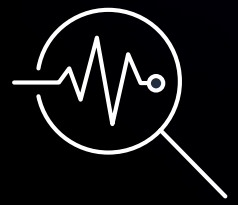
Globe Life (NYSE: GL) is headquartered in McKinney, TX, and has more than 14,000 insurance agents and 3,000 corporate employees.

With a mission to **Make Tomorrow Better**, Globe Life is the top volume issuer of ordinary individual life insurance policies in the U.S. and has more life insurance policyholders than any other insurance company.

More information is available at GlobeLifeInsurance.com.

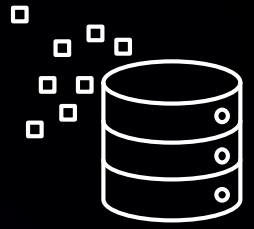
Source: S&P Global Market Intelligence; excluding reinsurance companies

Globe Life analytics objectives

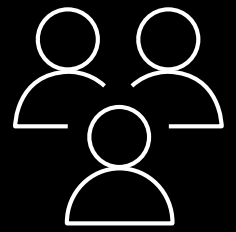


Enable self-service across business groups

Democratize data in a highly secure environment

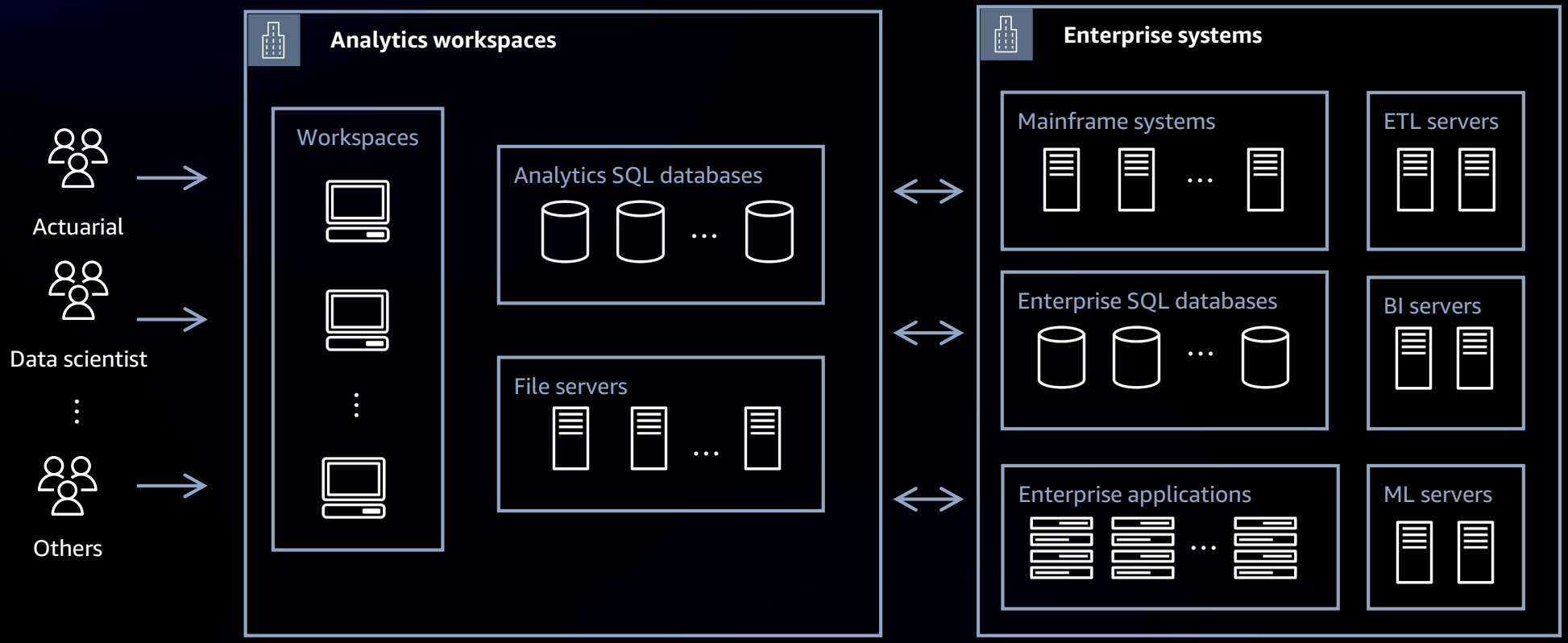


Modernize analytical applications



Allow IT to focus on value-added projects

Challenges with on-premises architecture



New compliance and regulatory requirements

Scaling complex actuarial models on growing data

Mainframe, legacy platforms lack flexibility

Lack of data sharing and collaboration

aws

Solution criteria

Separate compute/storage

Meet all security requirements

Scale to hundreds of users

Ease of use

Improved cost and performance

Solution on AWS

AWS components

User endpoints – Amazon WorkSpaces 

Data lake – Amazon S3, Amazon Athena, AWS Glue Data Catalog   

Relational database – Amazon Aurora PostgreSQL 

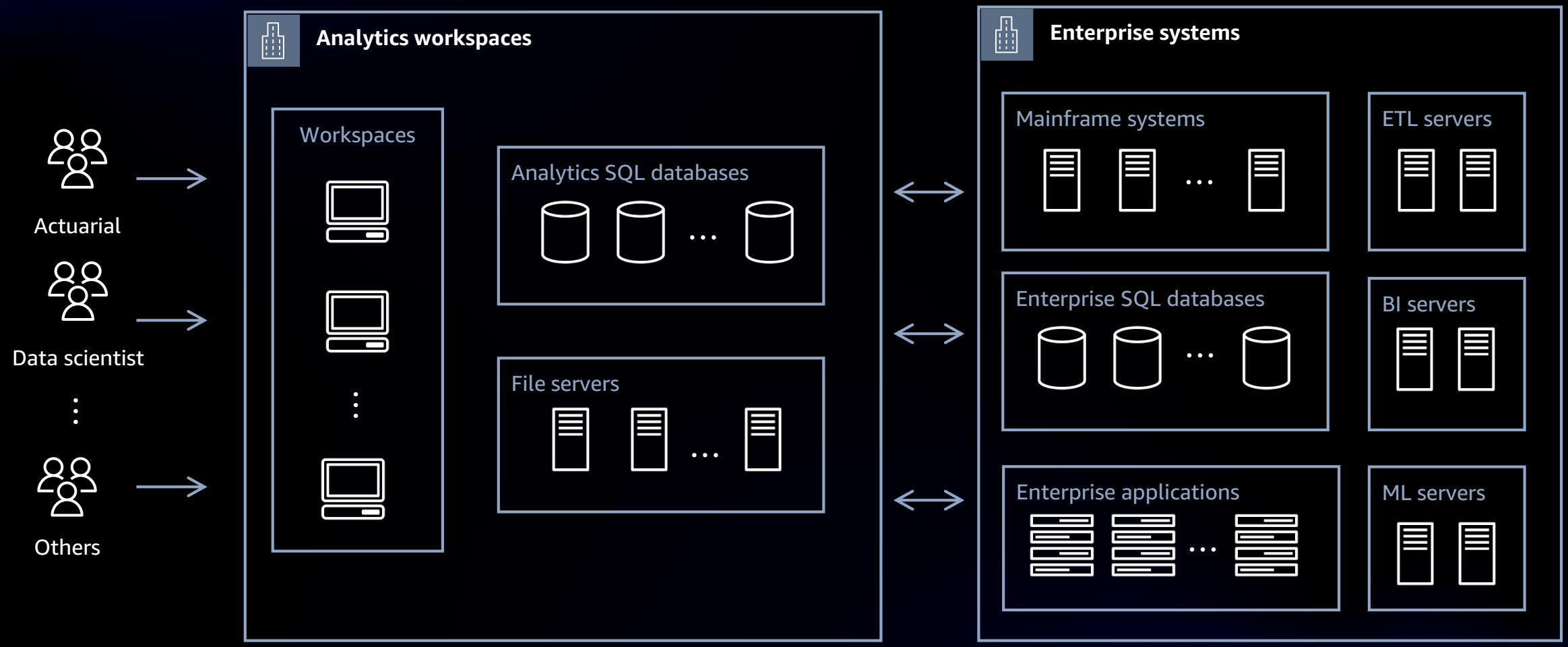
Data warehouse – Amazon Redshift 

Compute – Amazon EC2, AWS Lambda, AWS Glue   

Orchestration – AWS Step Functions, AWS Systems Manager, Amazon MWAA   

aws

On-premises architecture



High-level AWS architecture

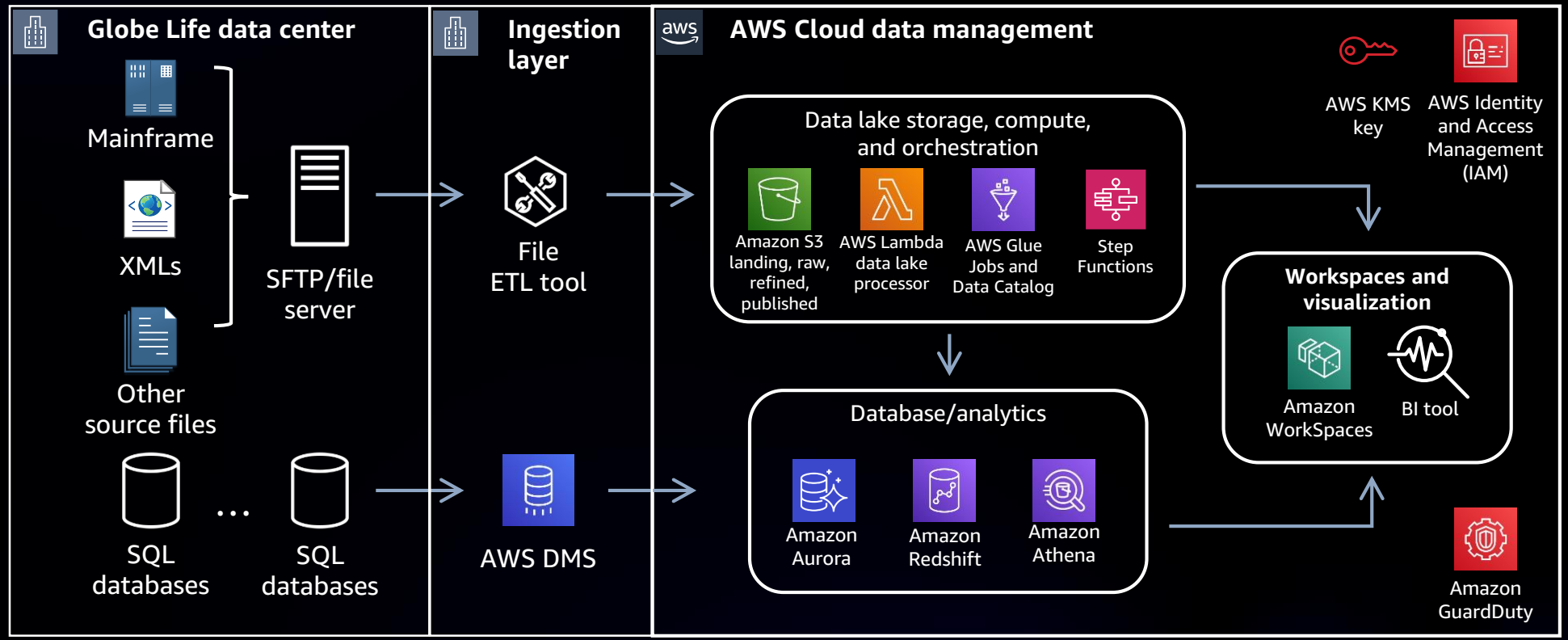
Build foundation

Ingest raw data

Serverless processing

Build data catalog

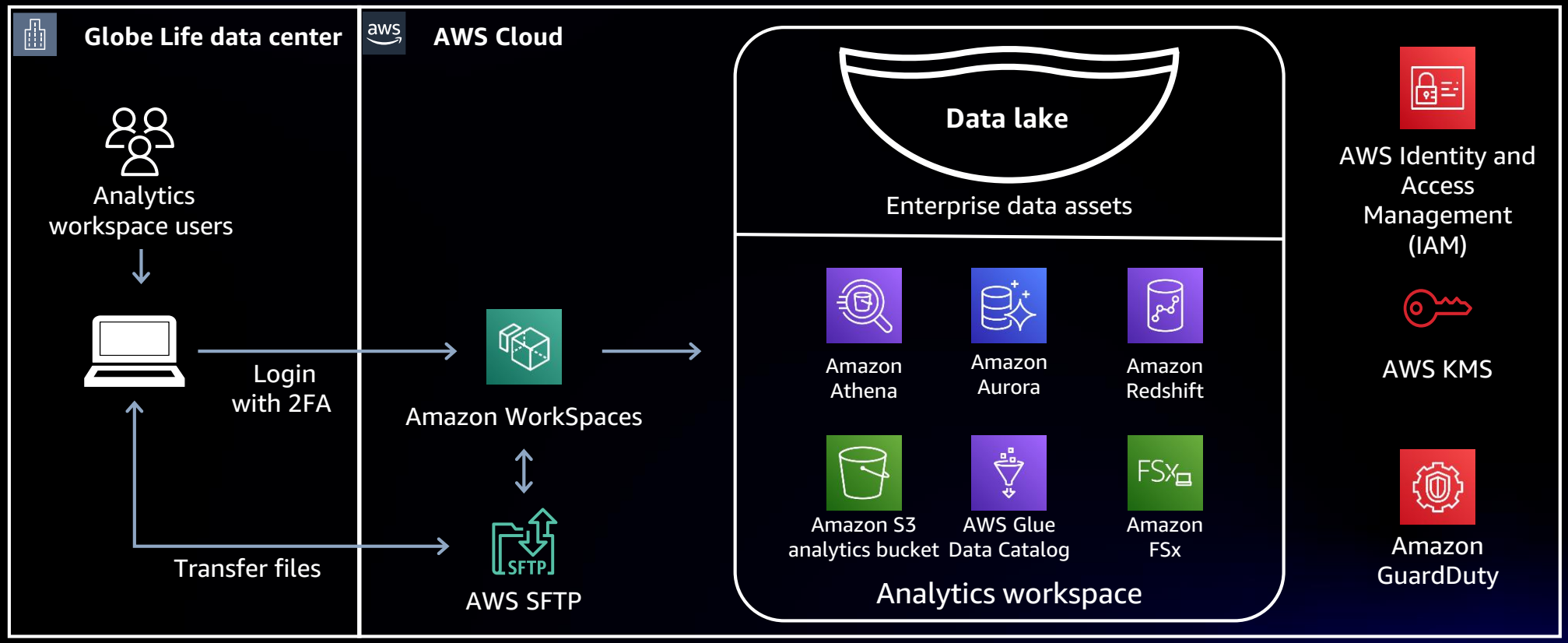
Enable analytics



Analytics workspaces

Why Amazon WorkSpaces?

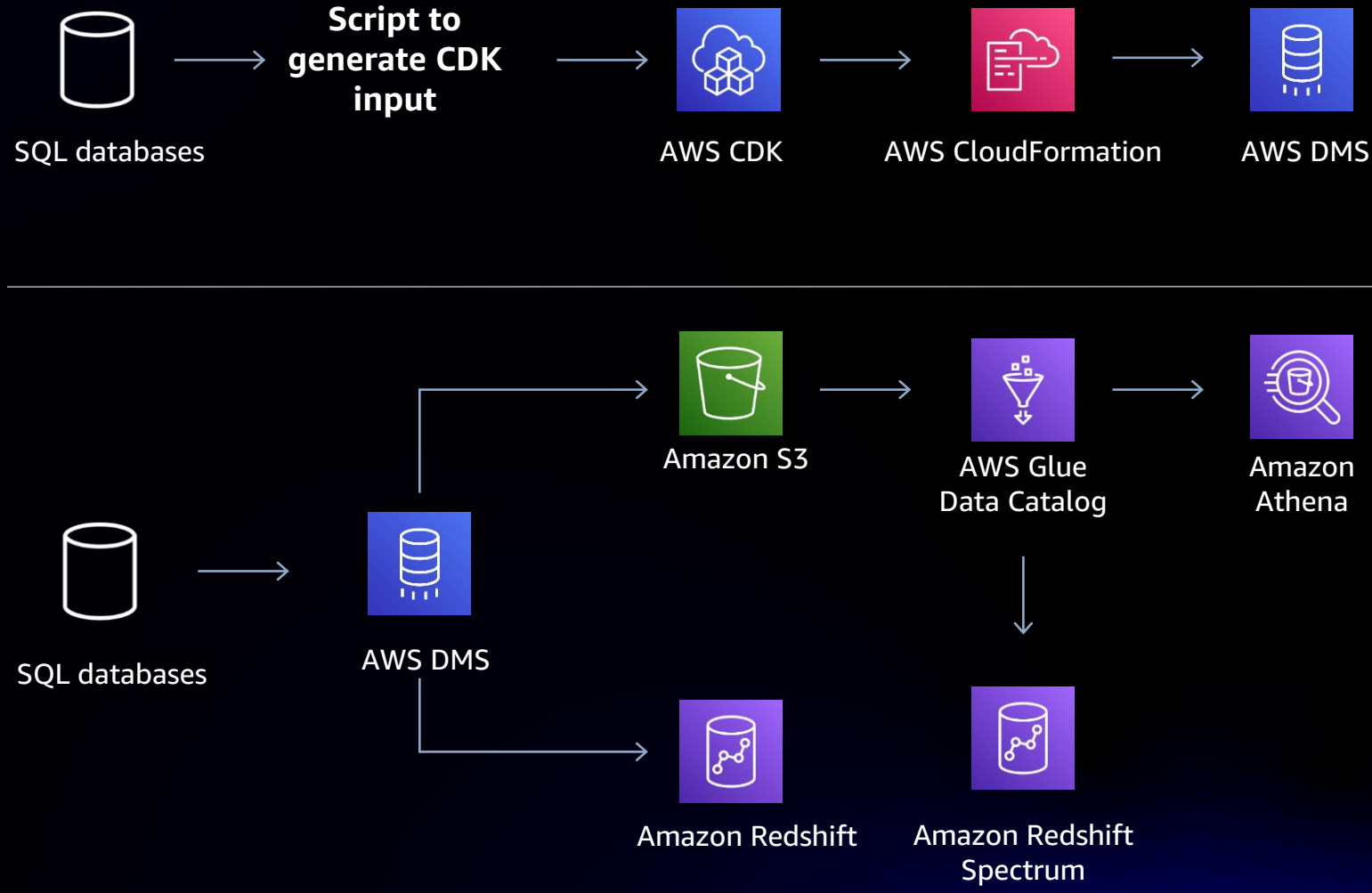
1. Secure cloud access
2. Scales up and down
3. Analytics tool compatibility



Data migration

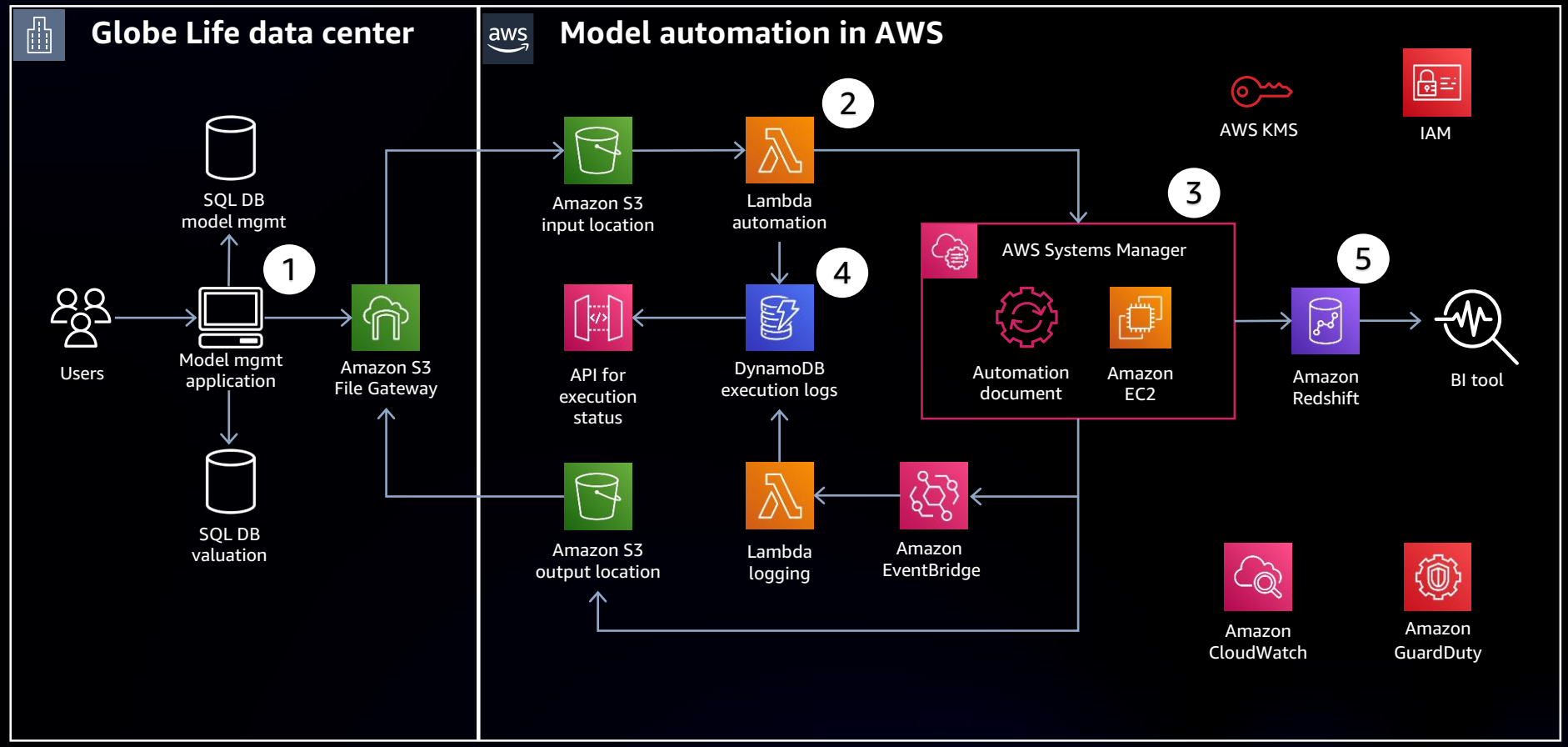
Hundreds of databases to be migrated with more than 100 TB of data

1. User provides inputs for data migration in specified template
2. Python program analyzes the DB/tables and generates input for AWS CDK
3. AWS CDK generates CFT for AWS DMS tasks for deployment
4. AWS DMS tasks execute to migrate data



Actuarial model execution

1. Model app to generate input for automation
2. AWS Lambda triggers automation doc creation
3. Systems Manager runs models on Amazon EC2
4. Status logged into Amazon DynamoDB
5. Output to Amazon Redshift for insights



Model automation sample scripts

```python
def handler(event, context=None, session=None):
    automation = PS_Automation(event=event, session=session)
    try:
        automation.initiate_lambda(automation)
    except PS_Error as error:
        PS_Logger.error(error)
    else:
        automation.details['AutomationExecutionId'] = automation.setup_automation()

    format_timestamps(automation.details)
    automation.details.update(automation.parameters())
    automation.session.s3.put_object(
        Bucket=automation.event.trigger_bucket(),
        Key=f"{automation.event.logs_path()}/{automation.event.job_id()}.details",
        Body=json.dumps(automation.details, indent=4, sort_keys=True)
    )
    PS_Logger.log("Lambda Execution Complete")
    return automation.details
```

1 Lambda handler

2 Validate model

```python
def initiate_lambda(self):
    self.validate_trigger_event()
    self.configuration = PS_Configuration(
        event=self.event,
        session=self.session,
    )
    self.ebs = PS_Ebs(
        volume_size=self.get_ebs_volume_size(),
        volume_type=self.ebs_volume_type,
        kms_key_id=self.session.kms_key_id()
    )
    self.details.update(self.configuration.to_dict())
    self.validate_model_path()
```

```python
def setup_automation(self):
    self.ssm_document = self.create_ssm_document()
    return self.start_automation()
```

3 Setup automation

Model automation sample scripts

Create SSM Automation doc 4

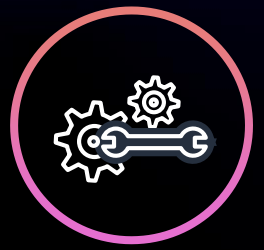
```python
def create_ssm_document(self):
    self.prepare_ssm_resources()
    self.check_ssm_document_existence()
    status = f"CreateSSMDocument::{self.document_name()}::"
    try:
        response = self.session.ssm.create_document(
            Content=json.dumps(self.document),
            Name=self.document_name(),
            DocumentType="Automation",
            DocumentFormat="JSON",
            Tags=self.tags()
        )
    except Exception as error:
        PS_Logger.log(f"{status}Failed")
        raise
    else:
        PS_Logger.log(f"Automation Document created:{self.document}")
        PS_Logger.log(f"{status}Success")
        return response
```

Start automation 5

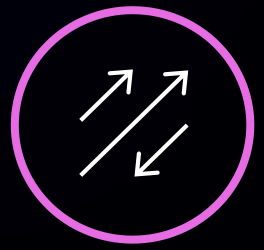
```python
def start_automation(self):
    try:
        automation_execution = self.session.ssm.start_automation_execution(
            DocumentName=self.ssm_document["DocumentDescription"]["Name"],
        )
    except (
        self.session.ssm.exceptions.AutomationDefinitionNotFoundException,
        self.session.ssm.exceptions.InvalidAutomationExecutionParametersException,
        self.session.ssm.exceptions.AutomationExecutionLimitExceededException,
        self.session.ssm.exceptions.AutomationDefinitionVersionNotFoundException,
        self.session.ssm.exceptions.IdempotentParameterMismatch,
        self.session.ssm.exceptions.InvalidTarget,
        self.session.ssm.exceptions.InternalServerError
    ) as error:
        PS_Logger.log(error)
        self.terminate_ec2_instance(self.details["InstanceId"])
        return
    else:
        self.log_document_to_s3()
        self.log_document_to_cloudwatch()
        self.log_details_to_cloudwatch()
        PS_Logger.log(f"{automation_execution['AutomationExecutionId']} by {self.event.config_file()}")
        return automation_execution['AutomationExecutionId']
```

Conclusion

Benefits achieved



Helps address regulatory requirements



Increased speed of innovation

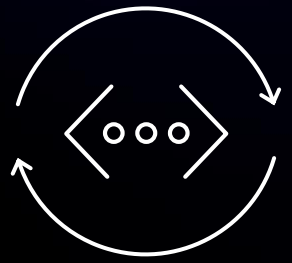
Enabled self-service

Solution achieved

- **Scalability** – Amazon WorkSpaces commissioned within minutes
- **Elasticity** – actuaries run models in parallel without IT
- **Performance** – up to 91% improvement with Amazon Redshift
- **Cost control** – ability to pause Amazon WorkSpaces, Amazon Redshift, and Amazon Aurora
- **Federation** – query across Amazon Redshift, data lake, and Amazon Aurora

aws

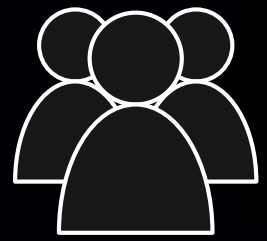
Lessons learned



Planning is critical

Involve AWS early

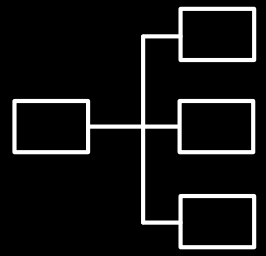
Define success criteria



Human element

Build trust with business

Understand current and required skills



Self-service is evolving

Establish mechanisms to manage security and data sharing

Moving forward

Access control and governance

AWS Lake Formation

Machine learning models in AWS

Amazon SageMaker

Enable other advanced analytics groups

Manage data pipelines using Amazon MWAA

Amazon MWAA

aws

Thank you!