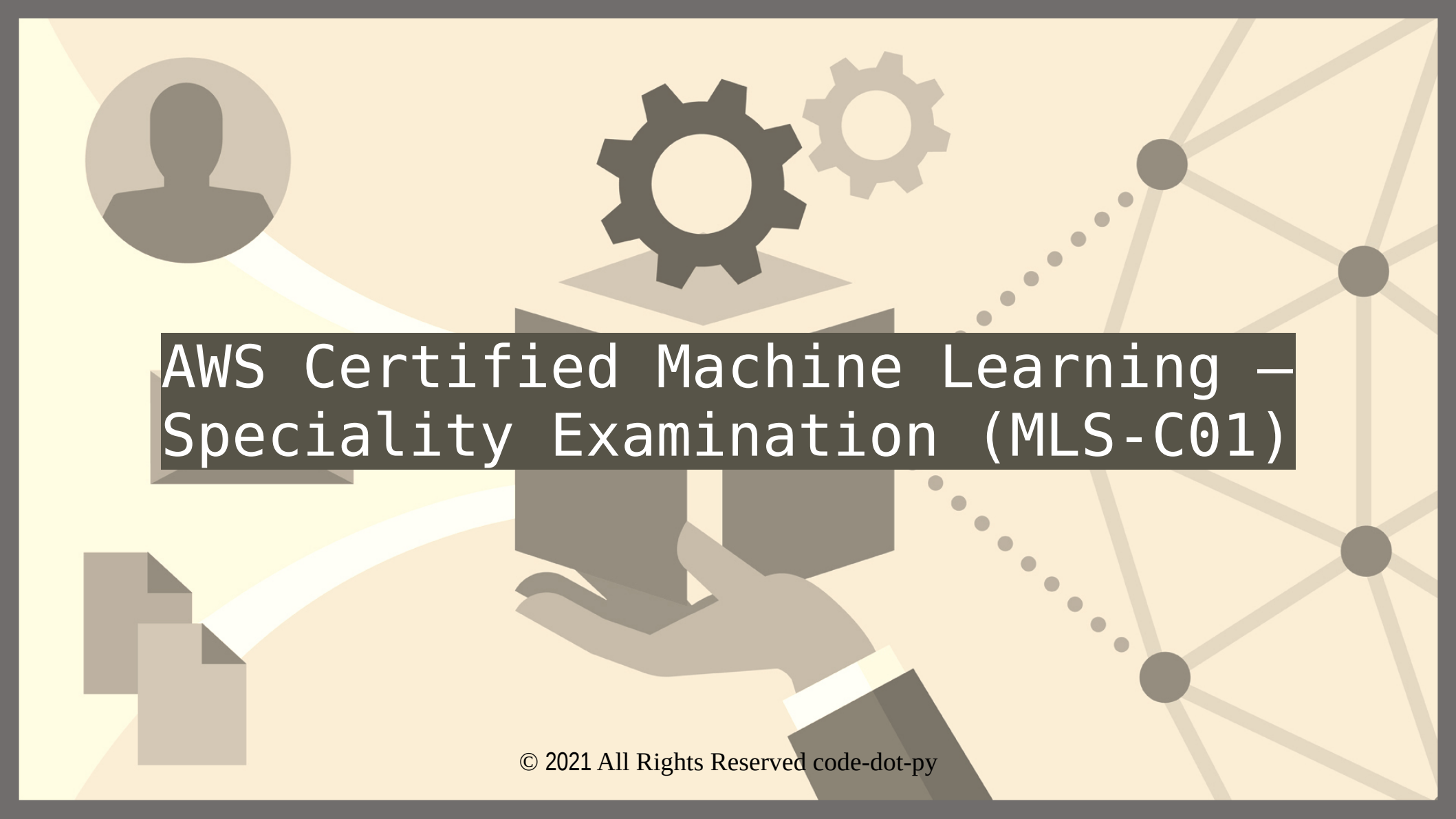
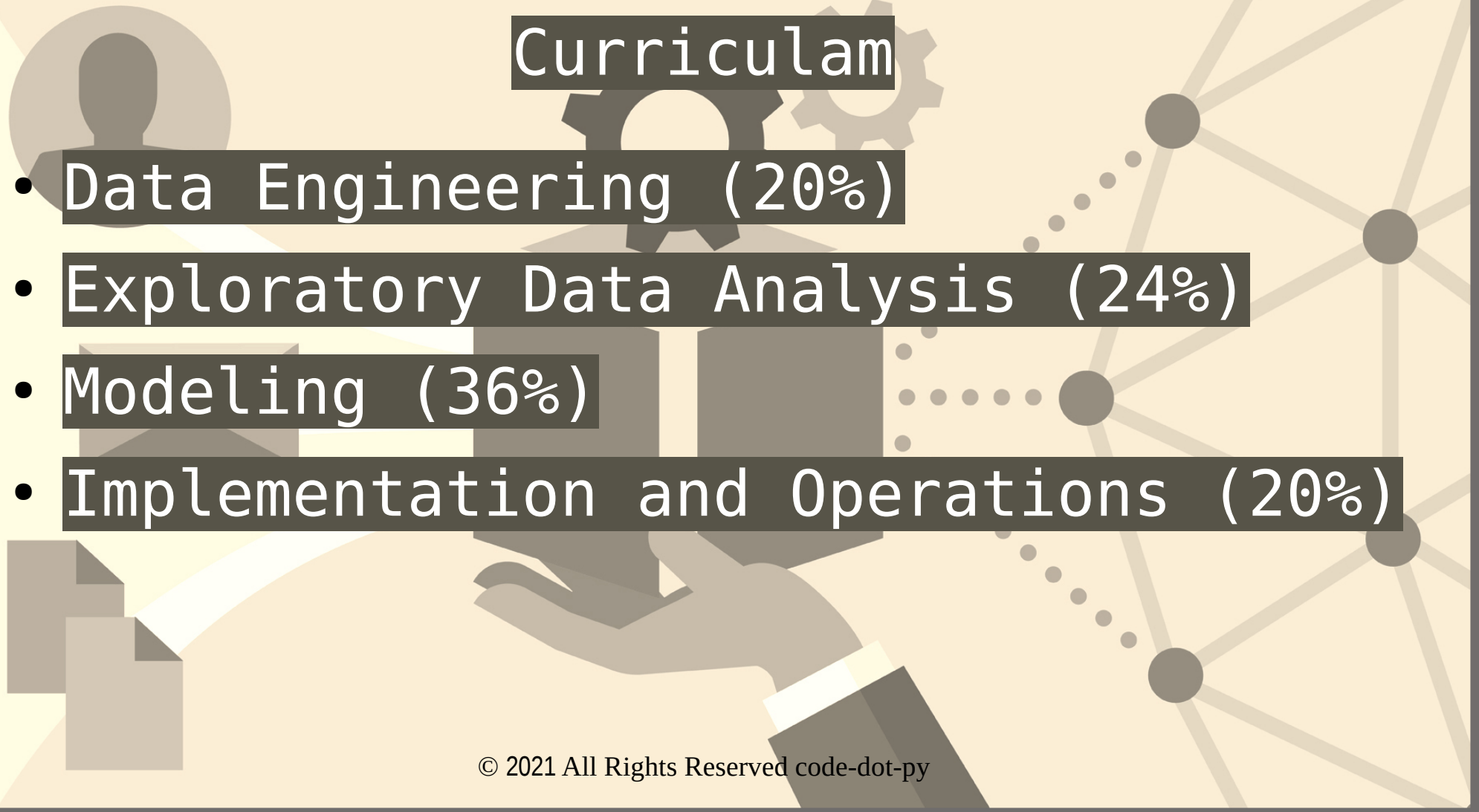


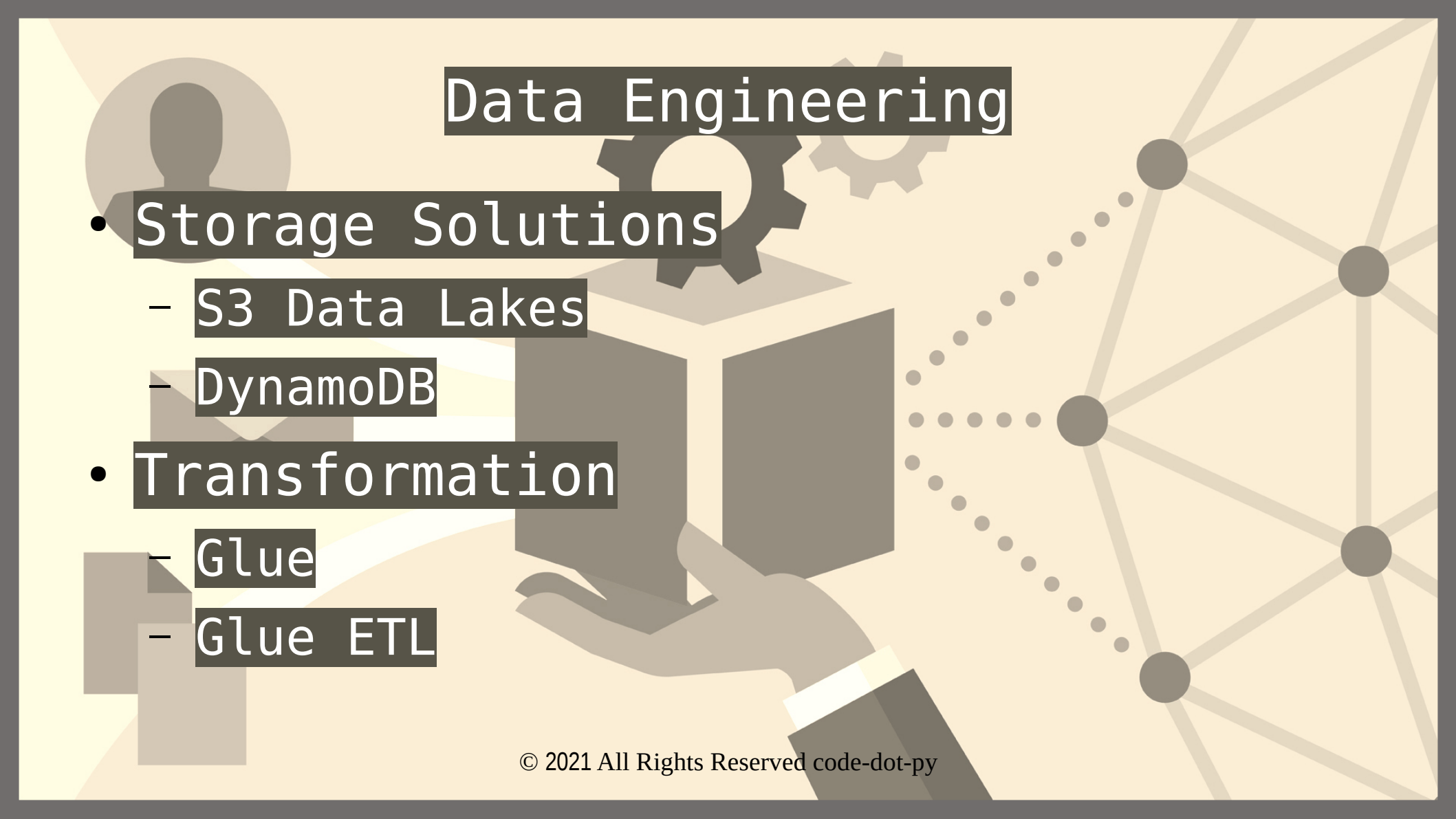
# AWS Certified Machine Learning – Speciality Examination (MLS-C01)

# Curriculam

- Data Engineering (20%)
- Exploratory Data Analysis (24%)
- Modeling (36%)
- Implementation and Operations (20%)

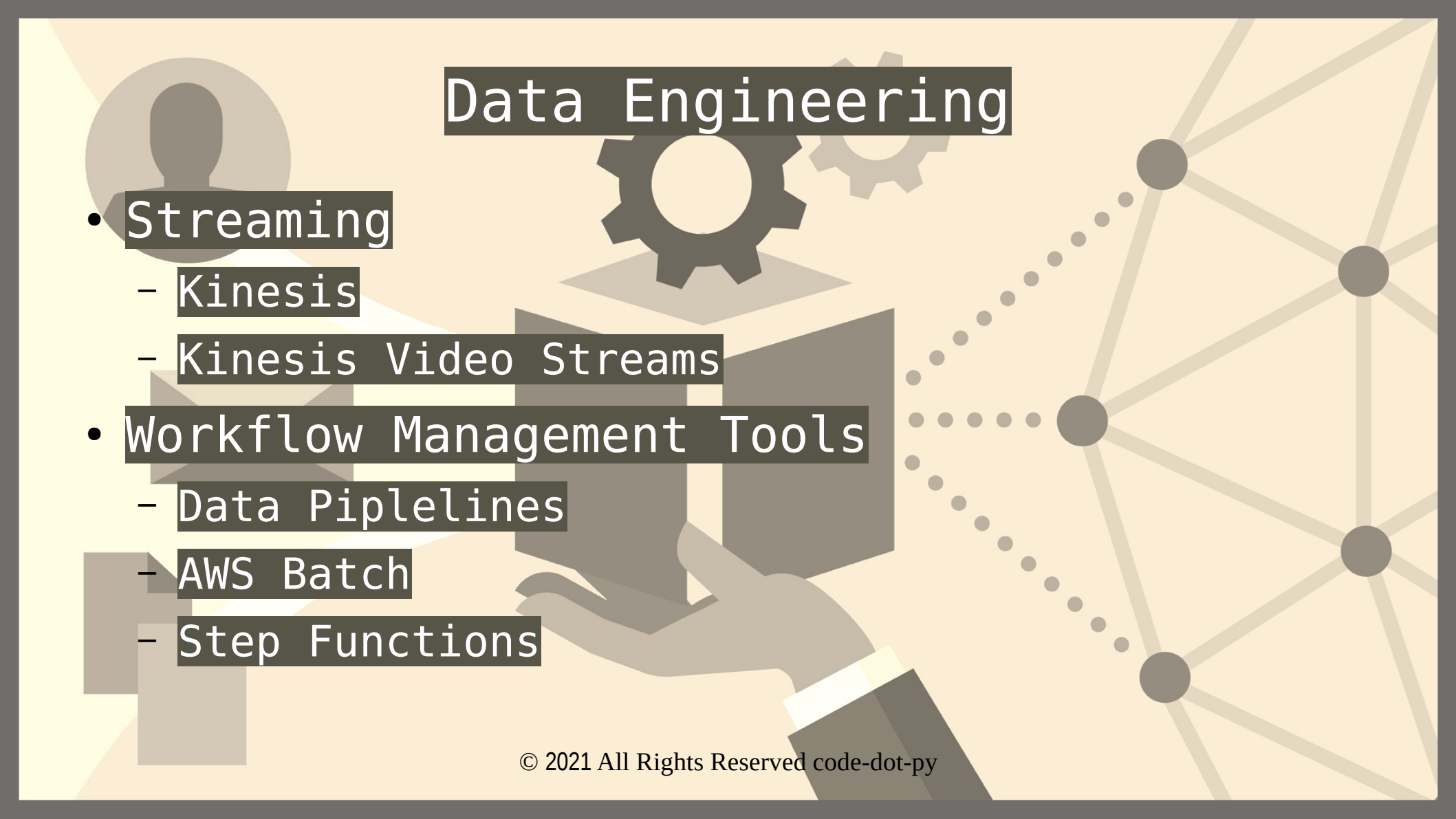


# Data Engineering



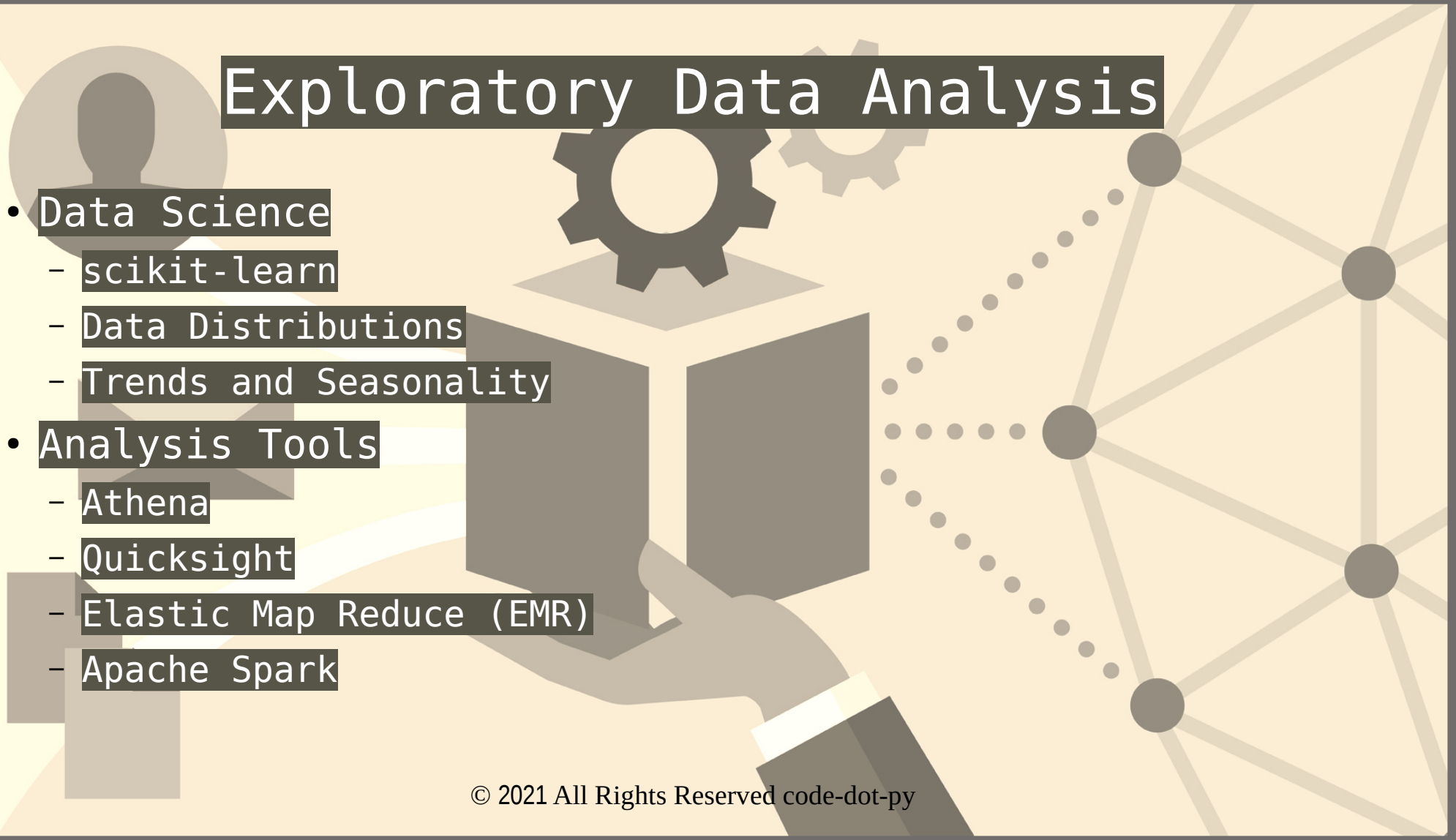
- Storage Solutions
  - S3 Data Lakes
  - DynamoDB
- Transformation
  - Glue
  - Glue ETL

# Data Engineering



- Streaming
  - Kinesis
  - Kinesis Video Streams
- Workflow Management Tools
  - Data Piplelines
  - AWS Batch
  - Step Functions

# Exploratory Data Analysis



- Data Science
  - scikit-learn
  - Data Distributions
  - Trends and Seasonality
- Analysis Tools
  - Athena
  - Quicksight
  - Elastic Map Reduce (EMR)
  - Apache Spark

# Exploratory Data Analysis

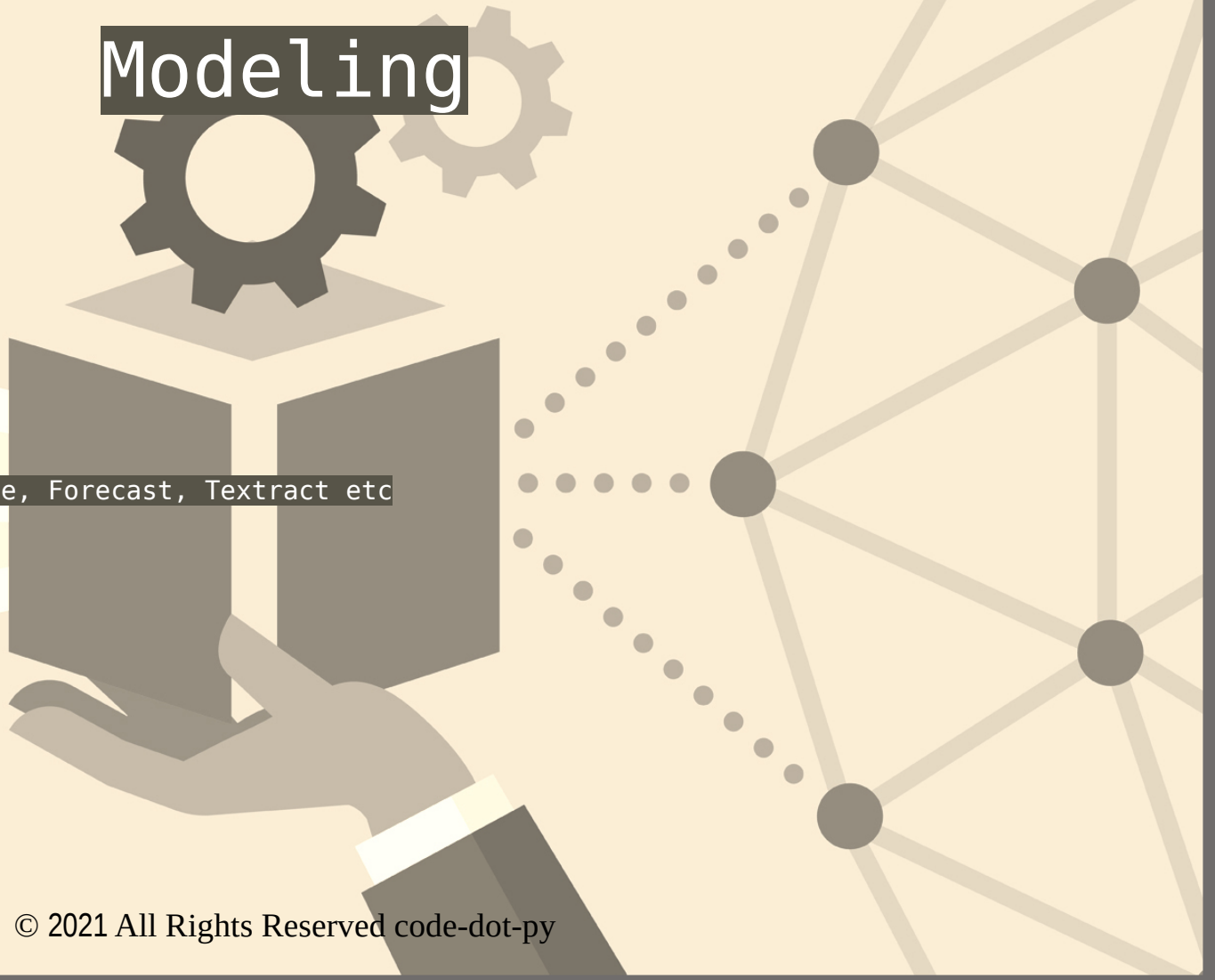
- Feature Engineering
  - Imputation methods
  - Outliers
  - Binning/Categorizing Data
  - Log transforms
  - One-hot encoding
  - Scaling and Normalization

# Modeling

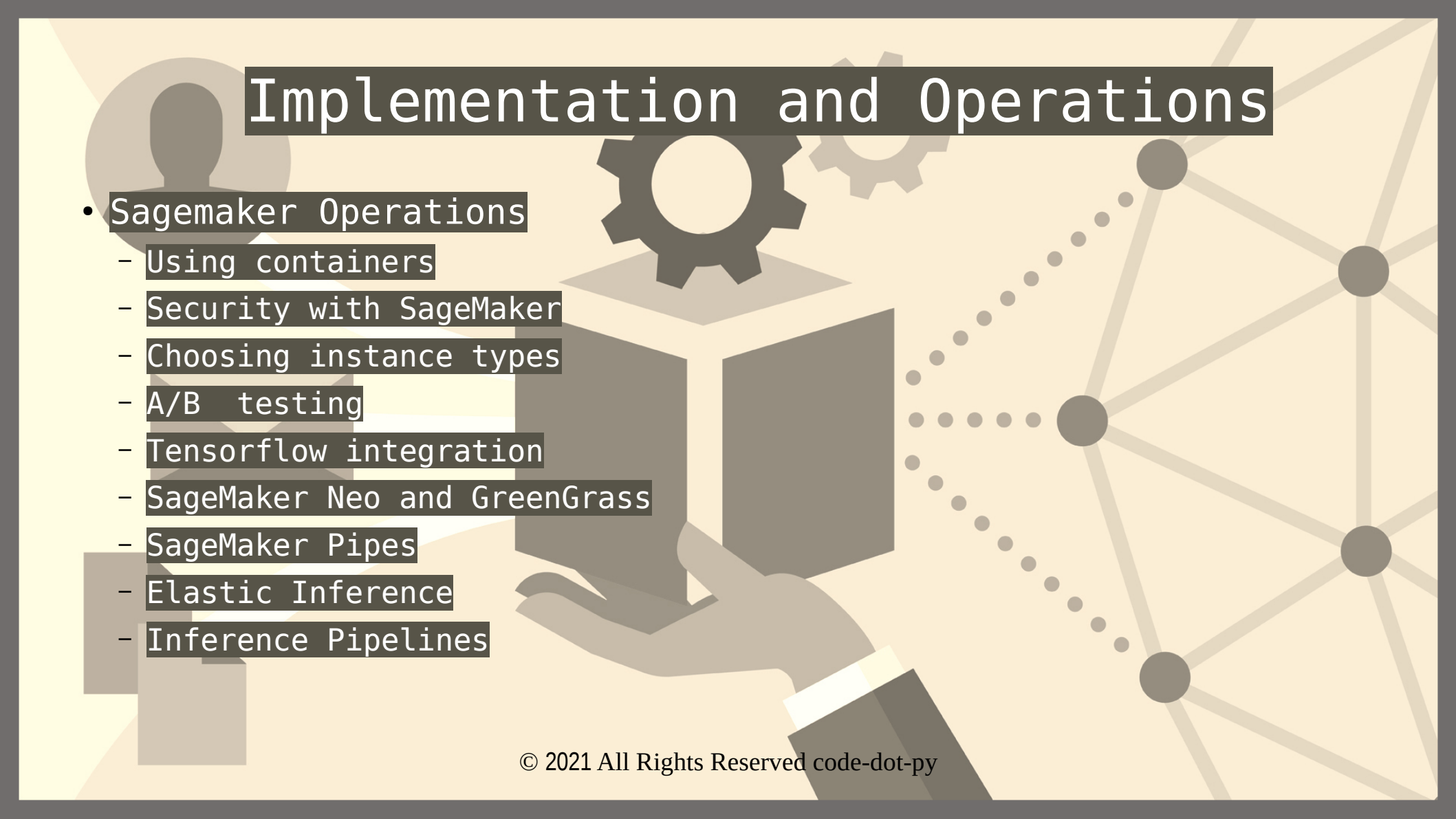
- Deep Learning
  - Multi-layer Perceptrons (MLPs)
  - Convolutional Neural Networks (CNNs)
  - Recurrent Neural Networks (RNNs)
  - ANN – Tuning and Regularization Techniques
- SageMaker
  - Architecture
  - Built-in Algorithms
  - Automatic Model Tuning
  - SageMaker Integration with other services - Spark

# Modeling

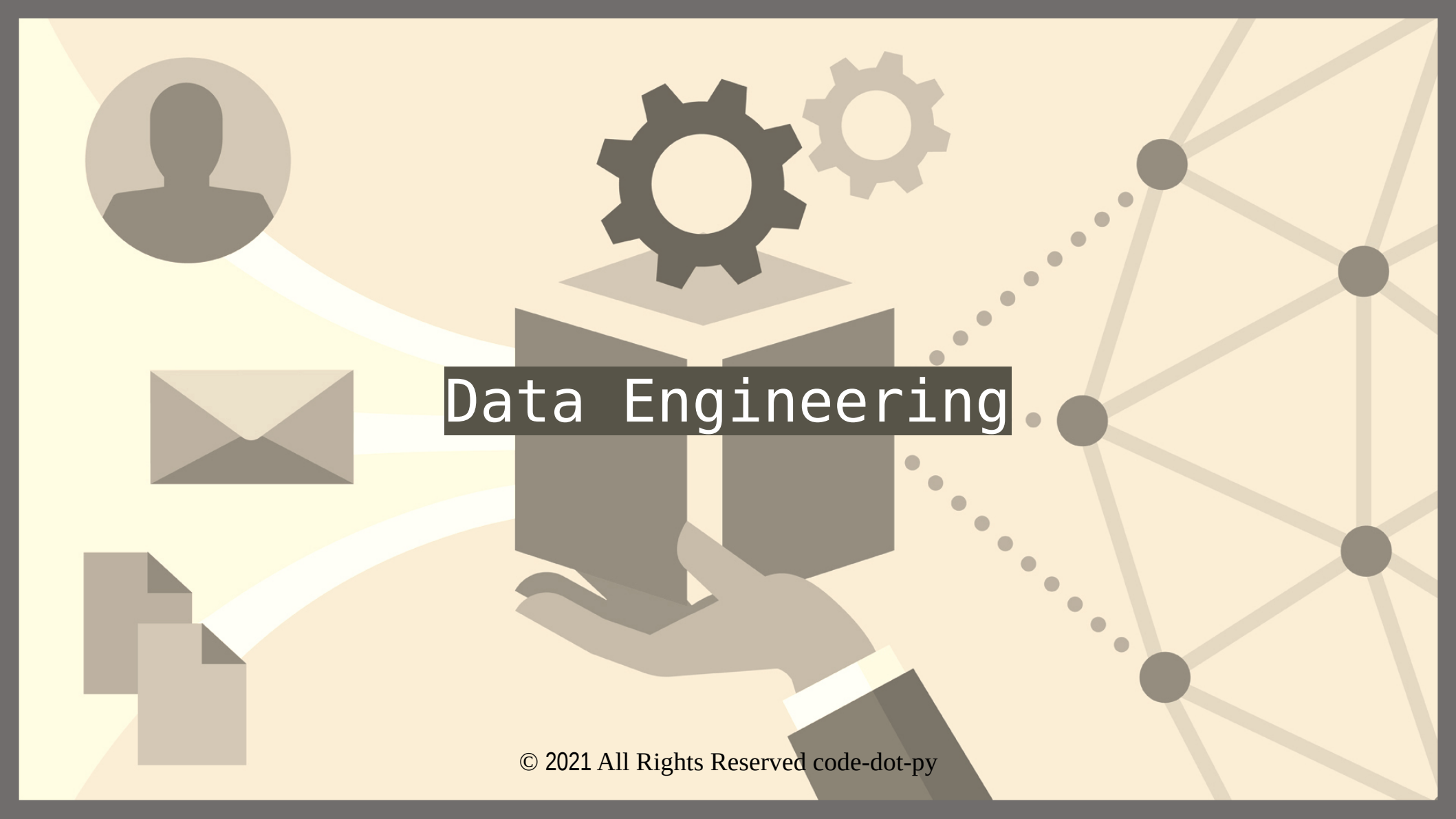
- High-level AI Services
  - Comprehend
  - Translate
  - Polly
  - Transcribe
  - Lex
  - Rekognition
  - Additional Services – Personalize, Forecast, Textract etc
  - DeepLens
- Evaluating and Tuning
  - Confusion Matrix
  - RMSE
  - Precision and Recall
  - F1 Score
  - ROC / AUC



# Implementation and Operations



- Sagemaker Operations
  - Using containers
  - Security with SageMaker
  - Choosing instance types
  - A/B  testing
  - Tensorflow integration
  - SageMaker Neo and GreenGrass
  - SageMaker Pipes
  - Elastic Inference
  - Inference Pipelines



# AWS S3 Overview

- S3 allows for storing objects (files) in buckets (directories)
- Buckets must have a globally unique name
- The full path of the objects is called ‘Key’. Example:
  - <bucketname>/<filename>.txt
  - <bucketname>/<foldername>/<filename>.txt
- The maximum object size that can be stored: 5TB

# AWS S3 for Machine Learning

- Backbone for many AWS ML services (Ex: SageMaker)
- Core service for Data Lake
  - Infinite size, no provisioning
  - 99.999999999% durability
  - S3 allows for decoupling (segregating) storage for all the compute based services. Examples:
    - EC2, Athena, Redshift, Rekognition, Glue
- Centralized Architecture – all the data at the same place
- Object Storage – supports any file format
- Common formats for ML – CSV, JSON, Parquet, ORC, Avro, Protobuf

# AWS S3 Data Partitioning

- Pattern for speeding up range queries (Eg: AWS Athena)
- Partitioning Examples:
  - By Date: s3://<bucketname>/<dataset>/year/month/day/hour/<datafile>.csv
  - By Product: s3://<bucketname>/<dataset>/product-id/<datafile>.csv
- We should choose the partitioning type based on use case
- Some tools like Kinesis and Glue can help with partitioning

# AWS S3 Storage Tiers

- Amazon S3 Standard – General Purpose (GP)
- Amazon S3 Standard – Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
  - Cheaper IA with diluted availability
- Amazon S3 Intelligent Tiering
  - New – Amazon determines where to put data to save cost
- Amazon Glacier
  - Archival

# AWS S3 Storage Tiers

| | Standard | Standard - Infrequent Access | One - Infrequent Access | S3 Intelligent-Tiering | Glacier |
|---|---|---|---|---|---|
| Durability | 99.999999999% | 99.999999999% | 99.999999999% | 99.999999999% | 99.999999999% |
| Availability | 99.99% | 99.9% | 99.5% | 99.90% | NA |
| AZ | ≥3 | ≥3 | 1 | ≥3 | ≥3 |
| Concurrent facility fault tolerance | 2 | 2 | 0 | 1 | 1 |

Frequently accessed | Infrequently accessed | Intelligent (new!) | Archives

# S3 Lifecycle Rules

- In order to save on cost, the lifecycle rules help in moving data between different tiers
- Example:
  - General Purpose (GP) -> Infrequent Access (IA) -> Glacier
- Transition actions – Objects are transitioned to another storage class
  - Move objects from:
    - GP to IA, 60 days post creation
    - IA to Glacier 6 months post creation
- Expiration actions – S3 deletes expired objects on our behalf
  - Log files can be set to delete after a specific period of time

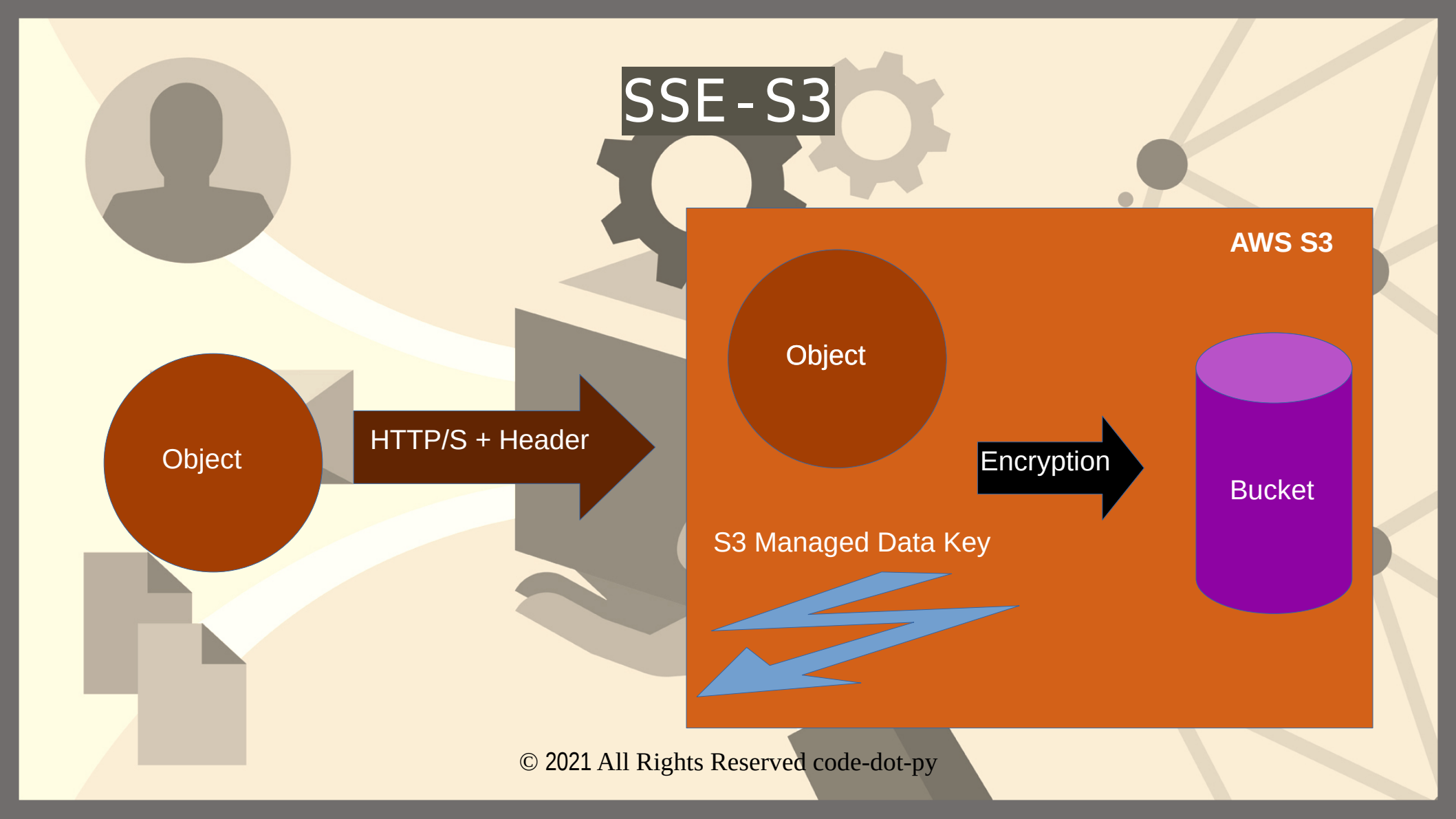
# S3 Security - Encryption for Objects

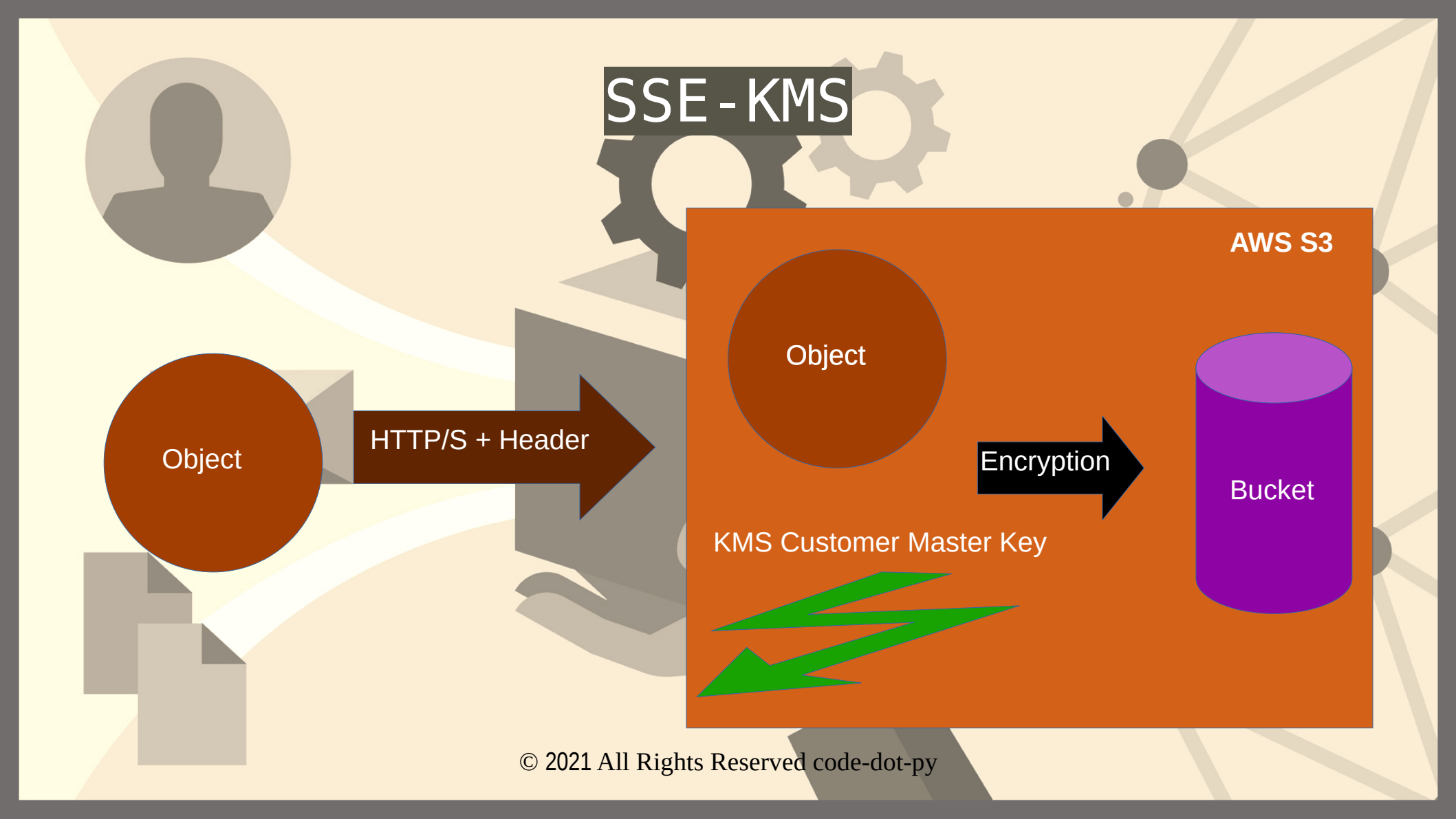
- There are four methods of encrypting objects in S3:

- SSE-S3: Encrypts S3 objects using keys handled and managed by AWS
- SSE-KMS: Use AWS key Management Service to manage encryption keys
  - Additional Security
  - Audit trail for KMS key usage

# S3 Security - Encryption for Objects

- SSE-C: We need to use our own encryption keys
- Client Side Encryption
- From an ML perspective, SSE-S3 and SSE-KMS will be the most likely used scenarios





# S3 Security

- User Based
  - IAM Policies – which API calls the user should be allowed
- Resource Based
  - Bucket Policy – allowing cross account access
  - Object Access Control List (ACL) – more precise control
  - Bucket Access Control List (ACL) – less commonly used

# S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Actions: Set of API to Allow or Deny
  - Effect: Allow / Deny
  - Principal: The account or user to apply the policy to

# S3 Bucket Policies

- Use S3 bucket policies for:
  - Granting public access to the bucket
  - For objects to be encrypted at upload
  - Grant access to another account (Cross Account)

# S3 Security – Points to Remember

- Networking – VPC Endpoint Gateway
  - Allow traffic to stay within your VPC
  - Make sure the private services (Eg: SageMaker) can access S3
- Logging and Audit:
  - S3 access logs can be stored in other S3 bucket
  - API calls can be logged in AWS CloudTrail
- Tagged Based (combined with IAM and bucket policies)

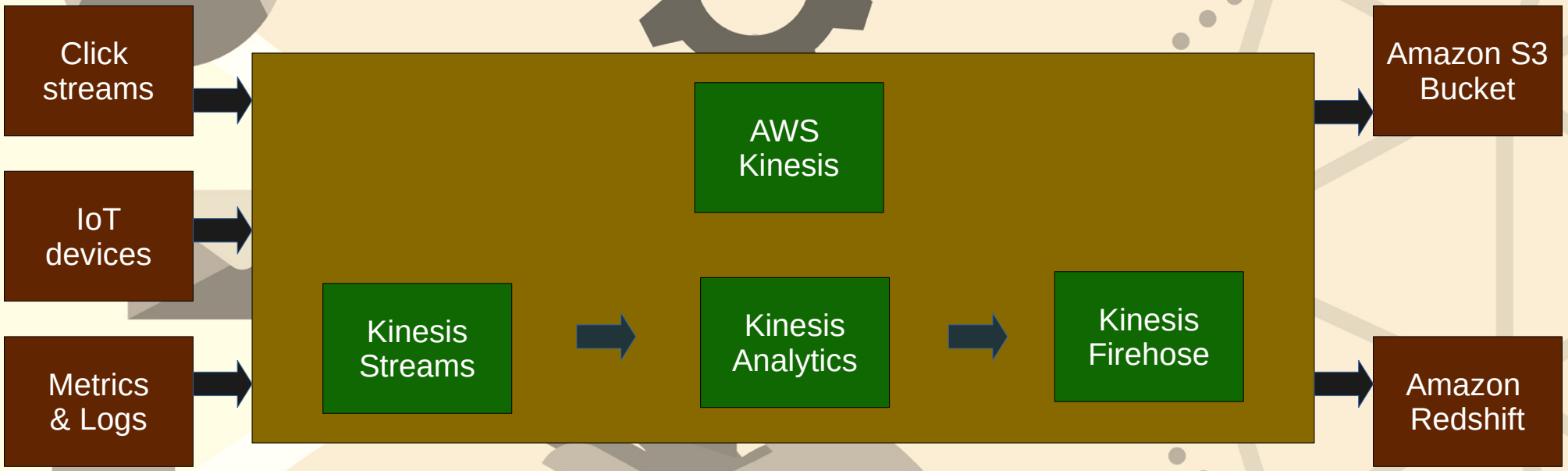
# AWS Kinesis - Overview

- Kinesis is a managed alternative to Apache Kafka
- Great for application logs, metrics, IoT, click streams etc
- Any reference to ‘real-time’ in the exam is an indication of relation to Kinesis
- Great for streaming processing frameworks (Spark, NiFi etc)
- Data is automatically replicated synchronously to 3 AZs

# AWS Kinesis – Key Services

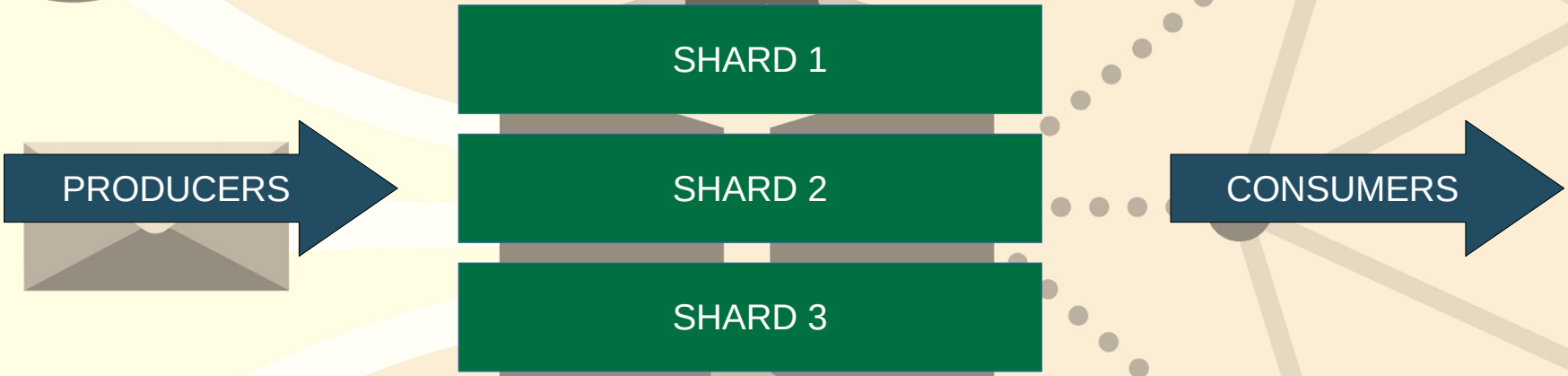
- Kinesis Streams – low latency streaming ingest at scale
- Kinesis Analytics – perform real-time analytics on streams using SQL
- Kinesis Firehose – load streams into S3, Redshift, ElastiSearch and Splunk
- Kinesis Video Streams – meant for streaming video in real-time

# AWS Kinesis – Architecture



# Kinesis Streams - Overview

- Streams are divided into ordered Shards / Partitions




- Shards have to be provisioned in advance (capacity planning)

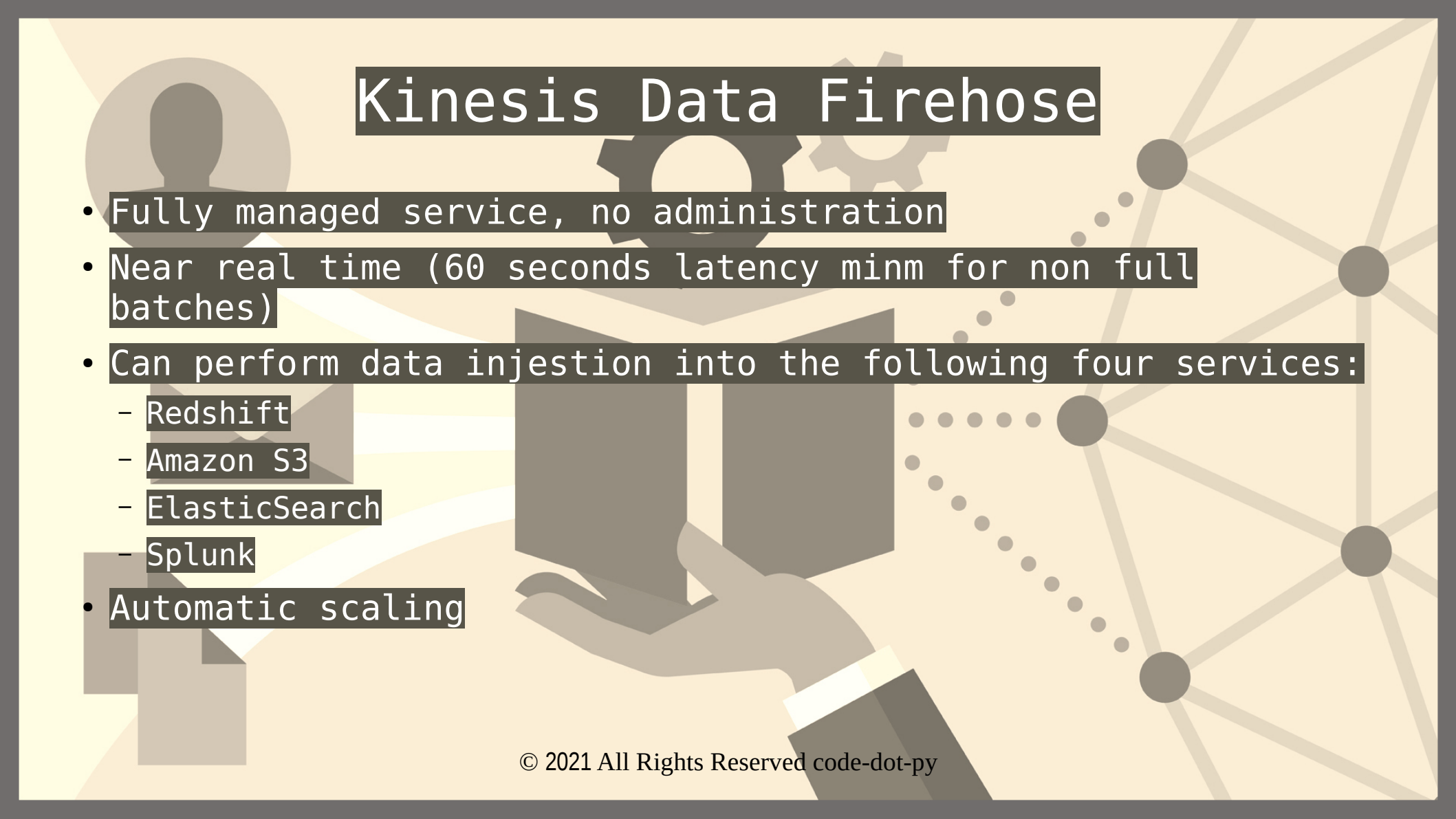
# Kinesis Streams - Overview

- Data retention is 24 hours by default and can go upto 7 days
- Ability to reprocess/replay data
- Multiple applications can consume the same stream
- Once data is inserted in Kinesis, it cannot be deleted (data immutability)
- Records can be upto 1 MB in size – fine for streaming use cases but not for large data analysis

# Kinesis Data Streams - Limits

- Producers:
  - 1MB/s or 1000 messages/s write speed per shard
  - If exceeded – 'ProvisionedThroughputException'
- Consumer (Classic):
  - 2MB/s read speed per shard across all consumers
  - Max 5 API calls/s/shard across all consumers
- Data Retention:
  - By default – 24 hours
  - Can be extended to upto 7 days

# Kinesis Data Firehose

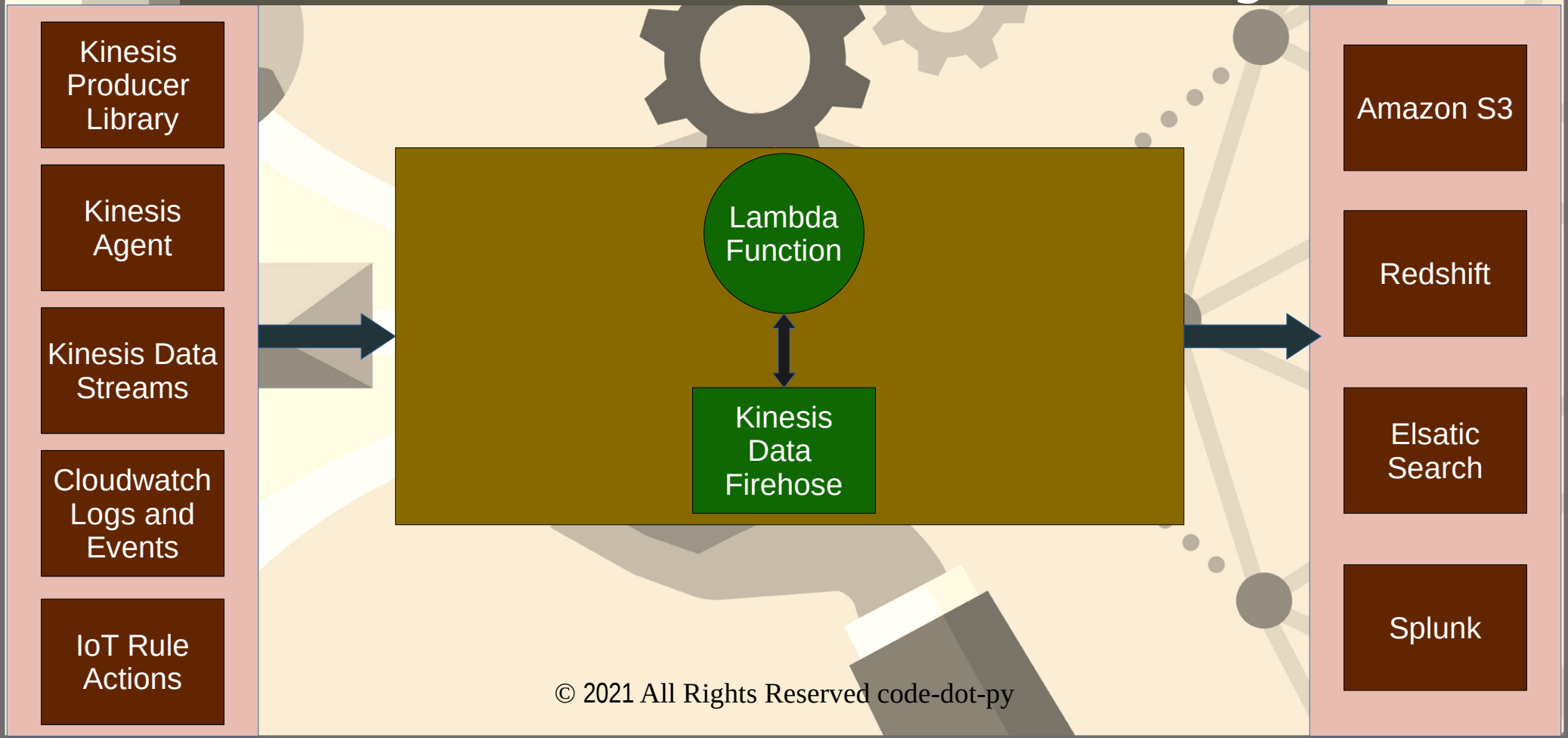


- Fully managed service, no administration
- Near real time (60 seconds latency minm for non full batches)
- Can perform data injestion into the following four services:
  - Redshift
  - Amazon S3
  - ElasticSearch
  - Splunk
- Automatic scaling

# Kinesis Data Firehose

- Supports many data formats
- Data conversion from CSV/JSON to Parquet/ORC (only for S3)
- Data transformation through AWS Lambda (CSV => JSON)
- Supports compression when target is Amazon S3 (GZIP, ZIP and SNAPPY)
- Pay for the amount of data going through Firehose

# Kinesis Data Firehose - Diagram



# Kinesis Data Streams vs Firehose

- Streams
  - Supports custom code writing for producer/consumer
  - Real-time  applications (latency ~200ms for classic and ~70ms for enhanced fan-out)
  - Must manage scaling ourselves (shard splitting/merging)
  - Data Storage for 1 to 7 days, replay capability, multi consumers

# Kinesis Data Streams vs Firehose

- Firehose
  - Deliver or ingestion service
  - Fully managed. Can send to S3, Splunk, Redshift, ElasticSearch
  - Serverless data transformations with Lambda
  - Near real time (lowest buffer time is 60 seconds)
  - Automated scaling
  - No data storage – no replay capability