AWS Secrets Manager

Table of Contents

What is AWS Secrets Manager?

1. 🔄 Manage, retrieve, and rotate secrets

📊 Various secret types

🔄 Throughout lifecycles

2. 🔐 Improve security posture

🚫 Eliminate hard-coded credentials

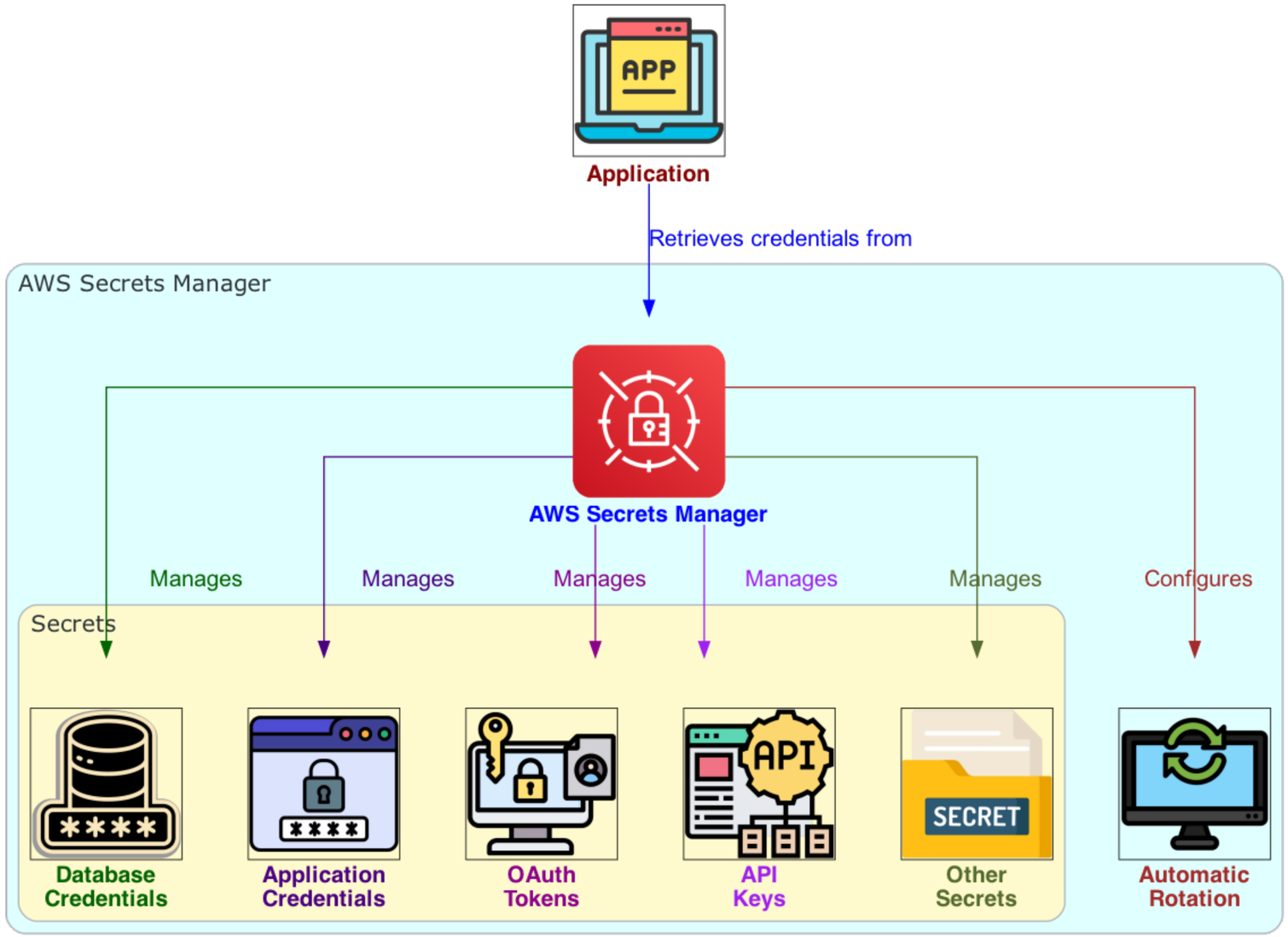
3. 🚫 Eliminate hard-coded credentials

🔒 Prevent potential compromise

4. 🏃 Dynamic runtime credential retrieval

🔍 Retrieve credentials dynamically

⏰ When needed



5. 🗓️ Automatic rotation schedule

📆 Configure rotation schedule

🔄 Regularly update, secure credentials

6. 🔄 Replace long-term with short-term secrets

🔄 Implement automatic rotation

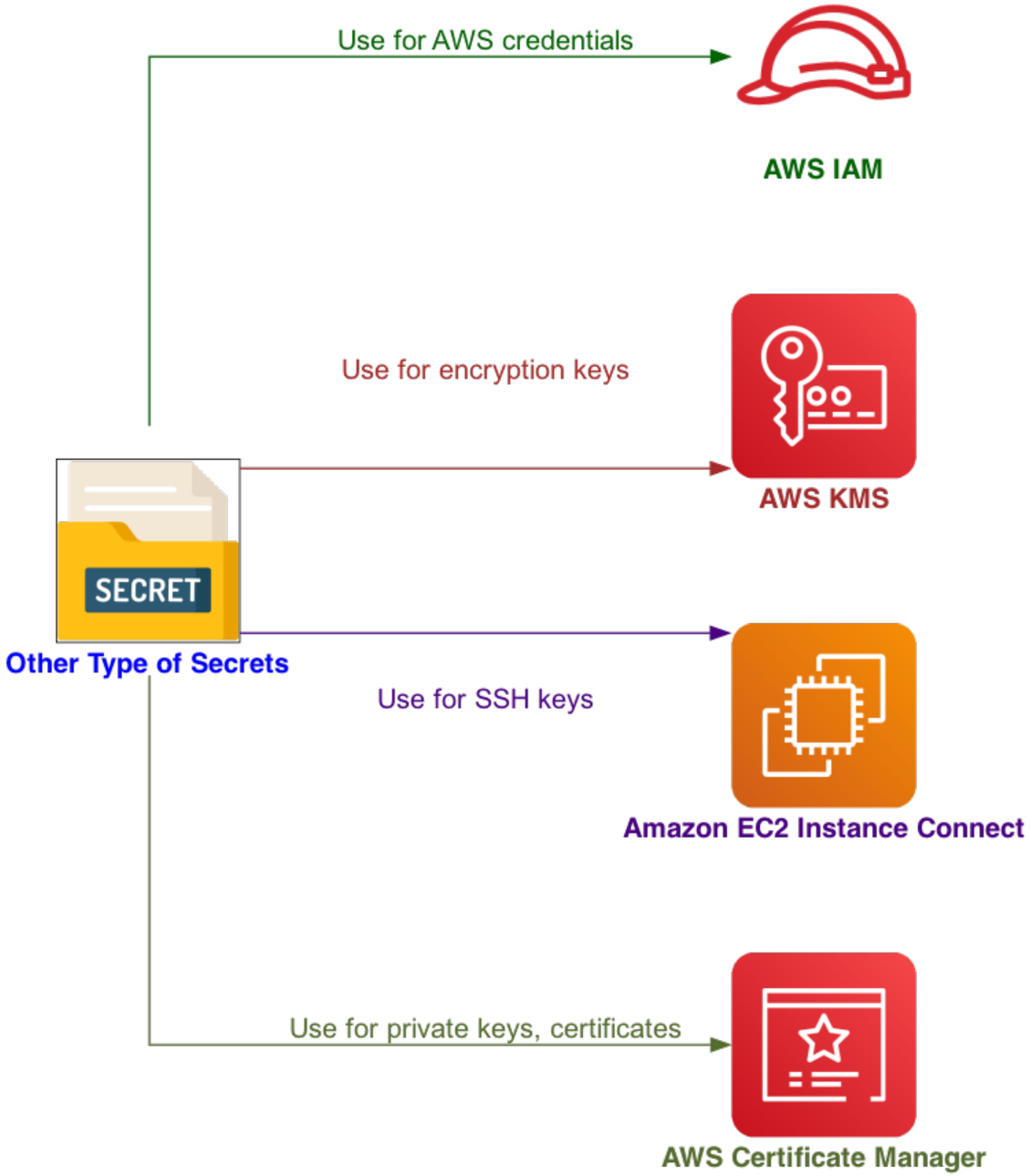
⬇️ Reduce risk of compromise

7. 🚀 Seamless credential rotation

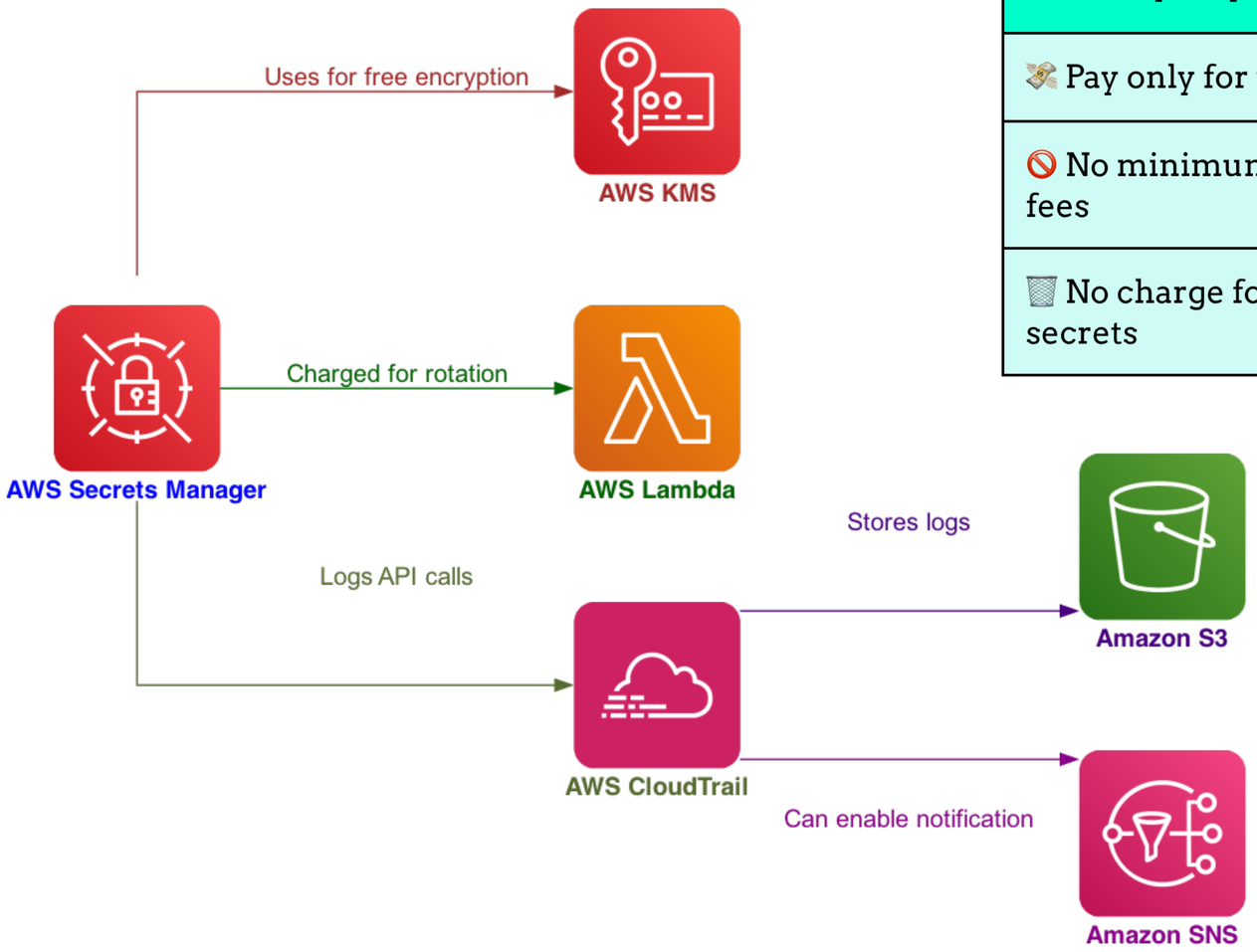
🚫 No application updates required

🚫 No client deployment changes needed

Other Type of Secrets



Pricing



1. 🔌 Pay-as-you-go model

- 💸 Pay only for usage
- 🚫 No minimum or setup fees
- 🗑️ No charge for deleted secrets

2. 🔑 Free AWS managed encryption key

- 🆓 Use `aws/secretsmanager` key for free
- 💳 KMS rate for custom keys

3. 🔄 Automatic rotation charges

- 🔃 Lambda used for rotation
- 💸 Charged at Lambda rate

4. 📝 CloudTrail logging costs

- 📋 Logs API calls
- 📝 Logs all events as management events

5. 💾 S3 storage fees for logs

- 🆓 First copy of management events free
- 💸 S3 charges for log storage

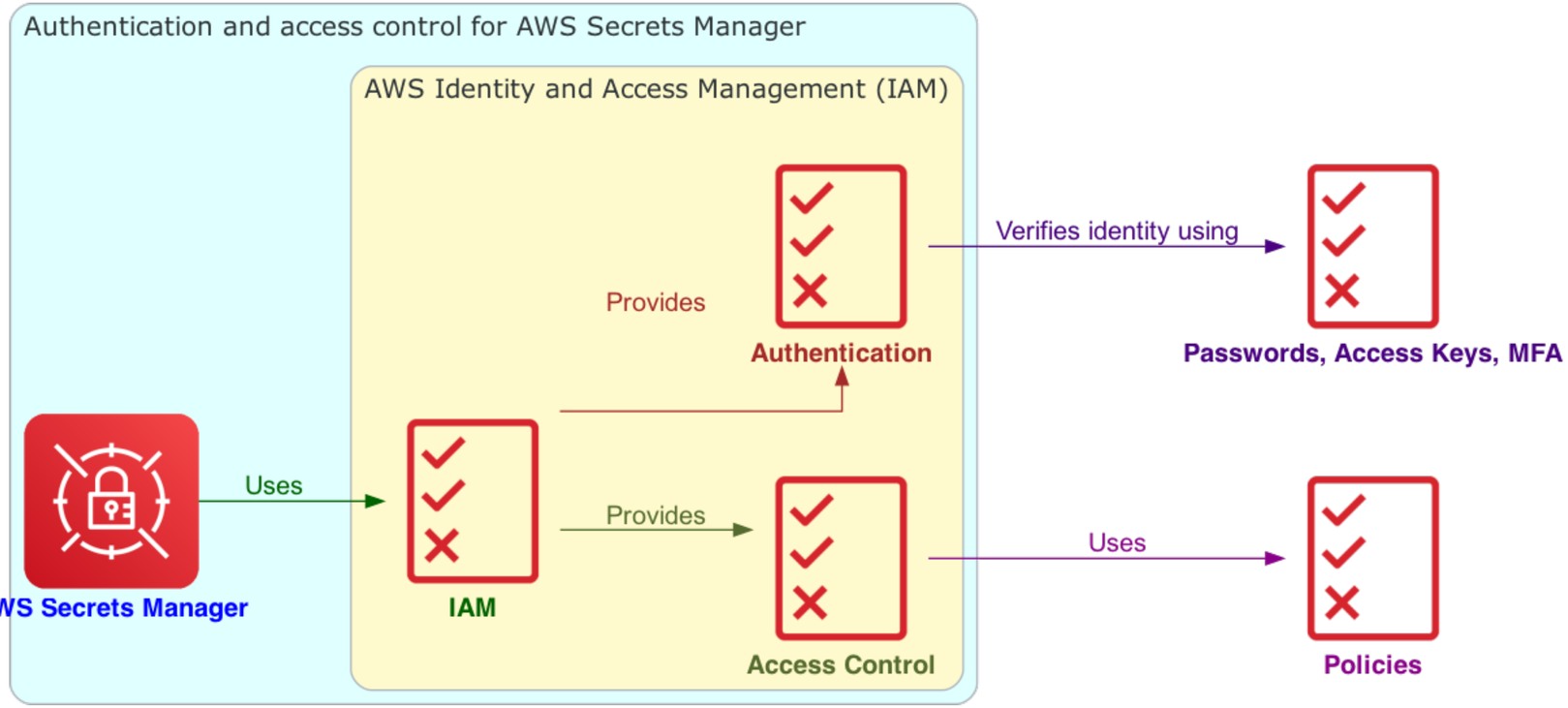
6. 📩 SNS notification charges

- 💸 Charges for enabling notifications

7. 🔍 Additional trail costs

- ➕ Multiple trails
- 💸 Extra costs for additional copies

🔒 Authentication and Access Control for AWS Secrets Manager 🗝️



4. 🚦 Access control for approved operations
✅ Ensures approved individuals only
🔒 Perform operations on AWS resources
🗝️ Secrets in Secrets Manager

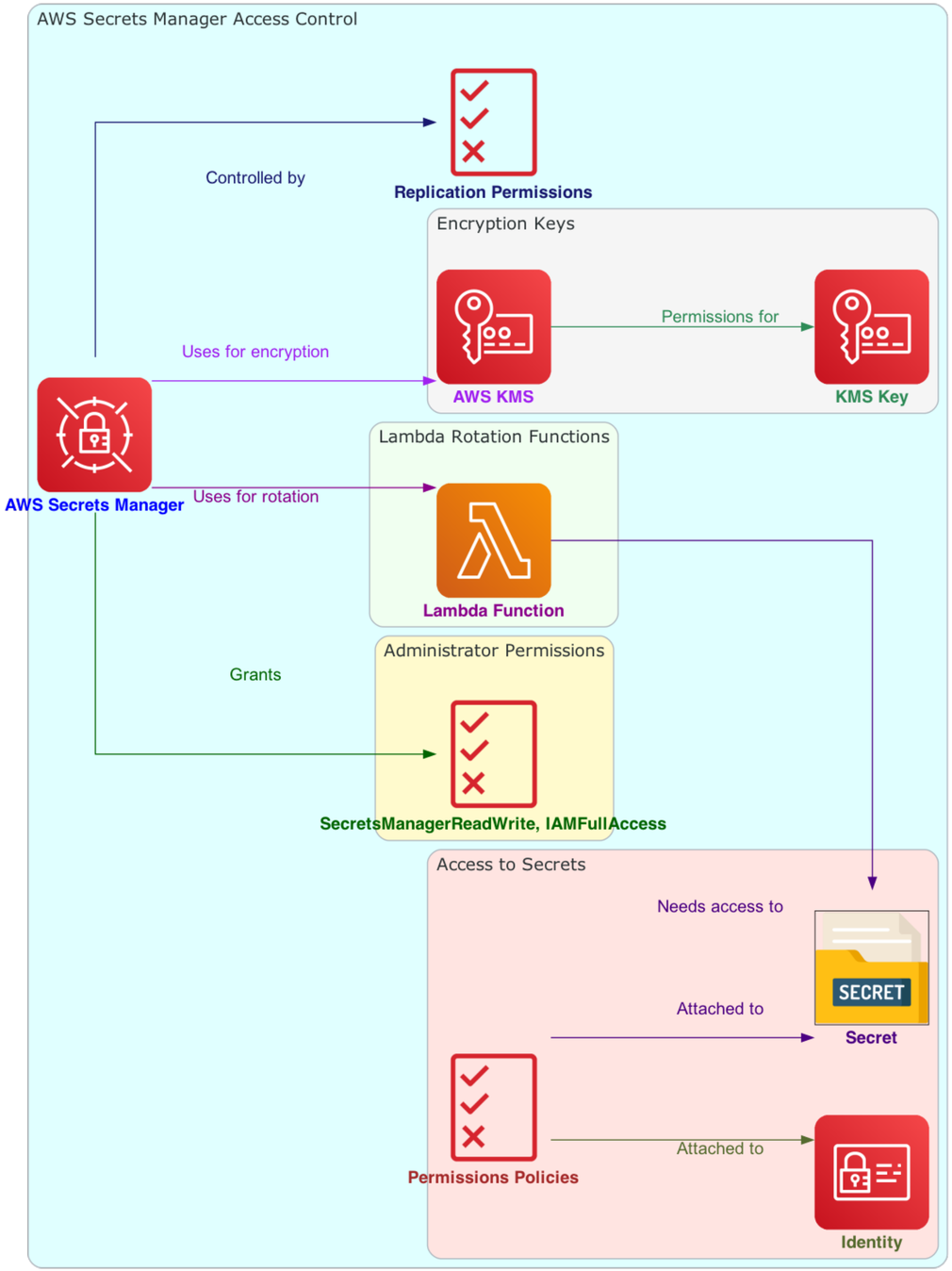
5. 🟫 Policies define access and actions
📋 Define who has access
🔒 Specific resources
🎭 Actions authenticated identity can take

1. 🔐 IAM secures access to secrets
🔒 Ensures secure access
🗝️ Secrets stored within service

2. 🔍 Authentication verifies identity
🆔 Verifies individuals' requests
🔐 Access to secrets in Secrets Manager

3. 🔑 Passwords, access keys, and MFA for authentication
🔐 Sign-in process
🔑 Passwords, access keys, MFA tokens
🆔 Verify identity of users

AWS Secrets Manager Access Control



👮 Secrets Manager administrator permissions
📜 SecretsManagerReadWrite
📜 IAMFullAccess

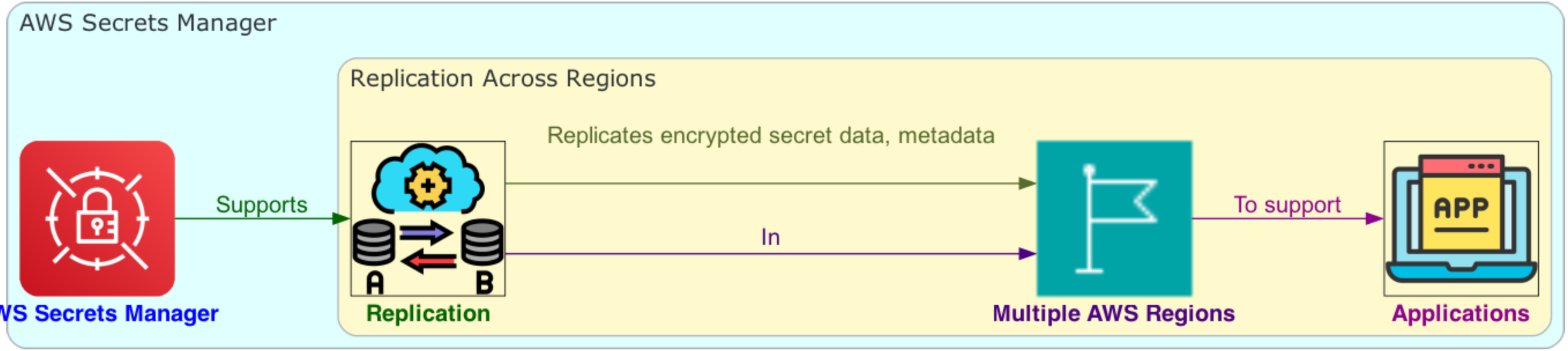
2. 🔑 Permissions to access secrets
📋 IAM permission policies
👤 Attach to identity
🔒 Attach to Secrets Manager secret

3. 🔄 Permissions for Lambda rotation functions
🔑 Access to secret
🗃️ Access to database or service

4. 🔐 Permissions for encryption keys
🗝️ AWS KMS keys
🆓 aws/secretsmanager key has permissions
🔑 Custom KMS key needs permissions

5. 🌐 Permissions for replication
📋 IAM permission policies
🌍 Control replication to other Regions

🌐 Replicating AWS Secrets Manager Secrets Across Regions 🔒



1. 🌍 Secrets replication in multiple AWS Regions

🗺️ Support applications across Regions

2. 🎯 Supports applications across Regions

🌍 Spread across different AWS Regions

3. ⏱️ Meets Regional access and low latency requirements

🔑 Regional access requirements

⚡ Low latency for applications

4. 🔄 Replica secrets can be promoted to standalone

🔒 Set up for independent replication

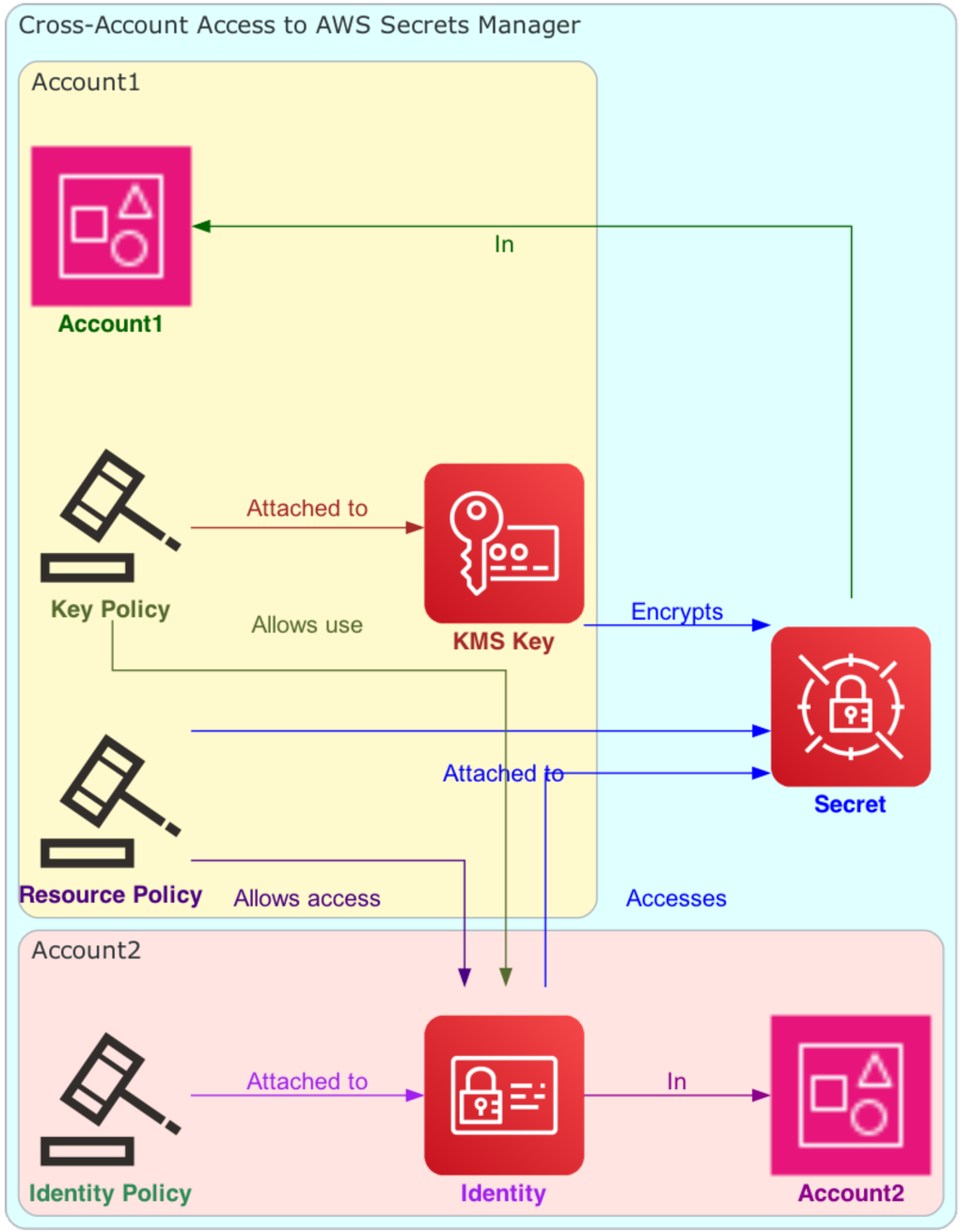
5. 🔐 Replicates encrypted secret data and metadata

🔑 Encrypted secret data

📋 Tags and resource policies

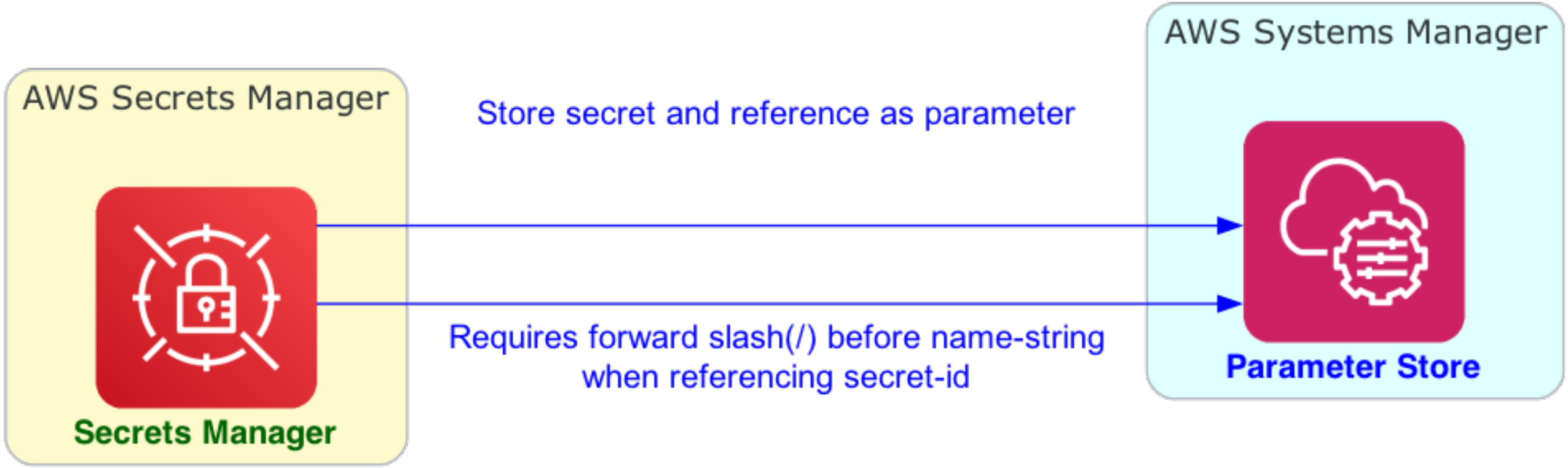
🗺️ Across specified Regions

🔑 Cross-Account Access to AWS Secrets Manager 🔒



1. 🔐 Allow access in resource and identity policies
   - 🔑 Resource policy
   - 👤 Identity policy
   - 🚫 Different from same account access
2. 🗝️ Use custom KMS key for encryption
   - 🚫 aws/secretsmanager not allowed
   - 🔐 Create custom KMS key
3. 💰 Charges apply for creating KMS keys
   - 💸 Required for cross-account access
4. 📜 Step 1: Attach resource policy to secret in Account1
   - 🔒 Account owning the secret
5. 🔑 Step 2: Add statement to KMS key policy in Account1
   - 🔓 Allow access from other account
6. 👤 Step 3: Attach identity policy in Account2
   - 🆔 Account needing access to secret

🔐 AWS Secrets Manager Integration with AWS Systems Manager Parameter Store



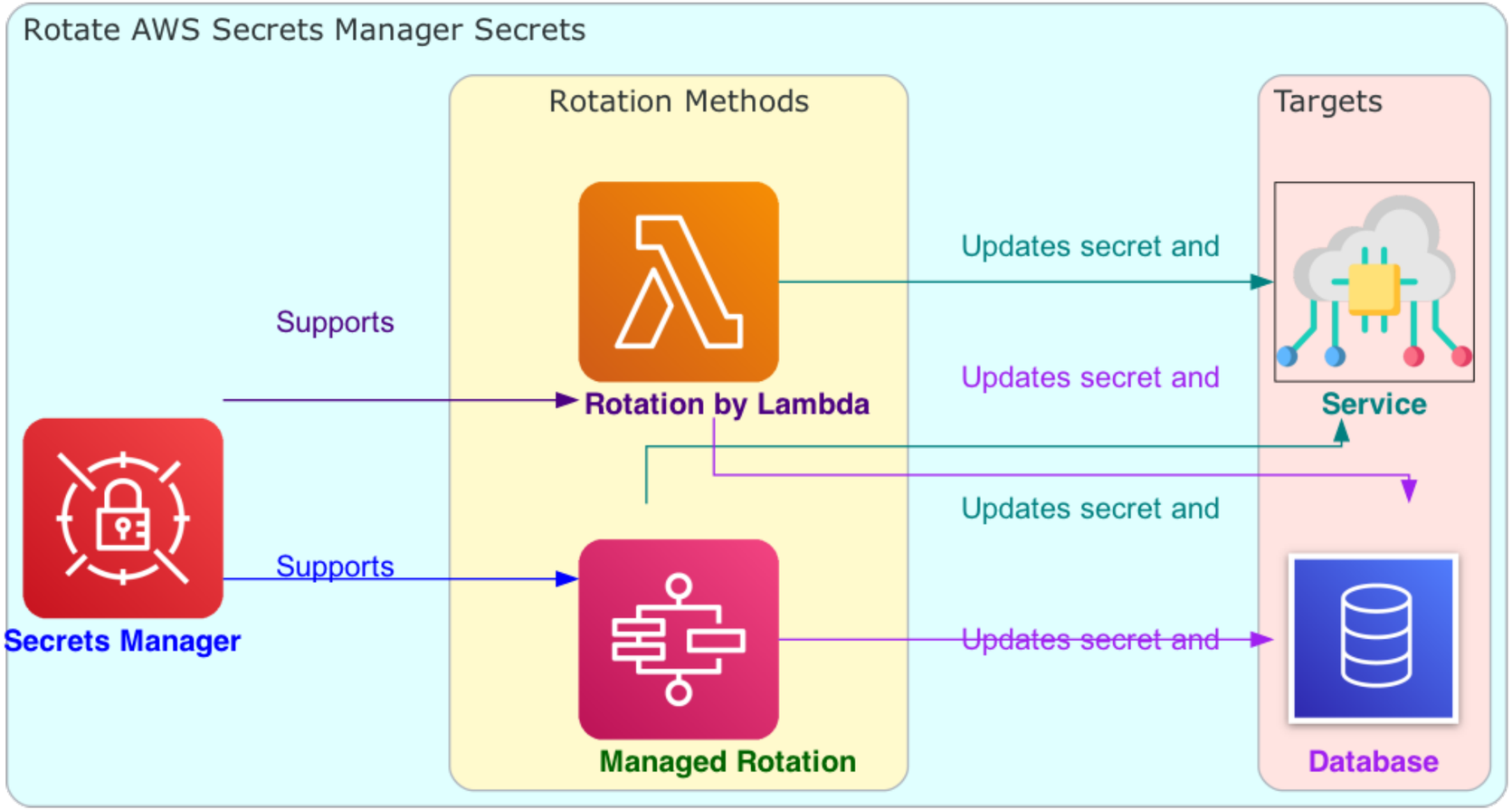
1. 🗄️ Parameter Store
🔒 Secure, hierarchical storage
⚙️ For configuration data, secrets
🔑 Passwords, database strings, license codes

2. 🔄 Secrets Manager
🔁 Automatic rotation services
🗝️ For stored secrets

3. 🔗 Reference Secrets Manager secrets as Parameter Store parameters
🌟 Leverage benefits of both services
🔐 Secure storage, automatic rotation
🗃️ Centralized configuration management system

4. 📝 Syntax for referencing secrets
🔑 secret-id requires (/) before name-string
🔀 Ensures proper integration

Rotate AWS Secrets Manager secrets



1. 📝 Definition

🔄 Periodically updating a secret

🔑 Update credentials in secret and database/service

⏰ Set up automatic rotation

2. 🔧 Two forms of rotation

🔋 Rotation by Lambda function

🔑 For other types of secrets

📜 Uses Lambda to update secret and database/service

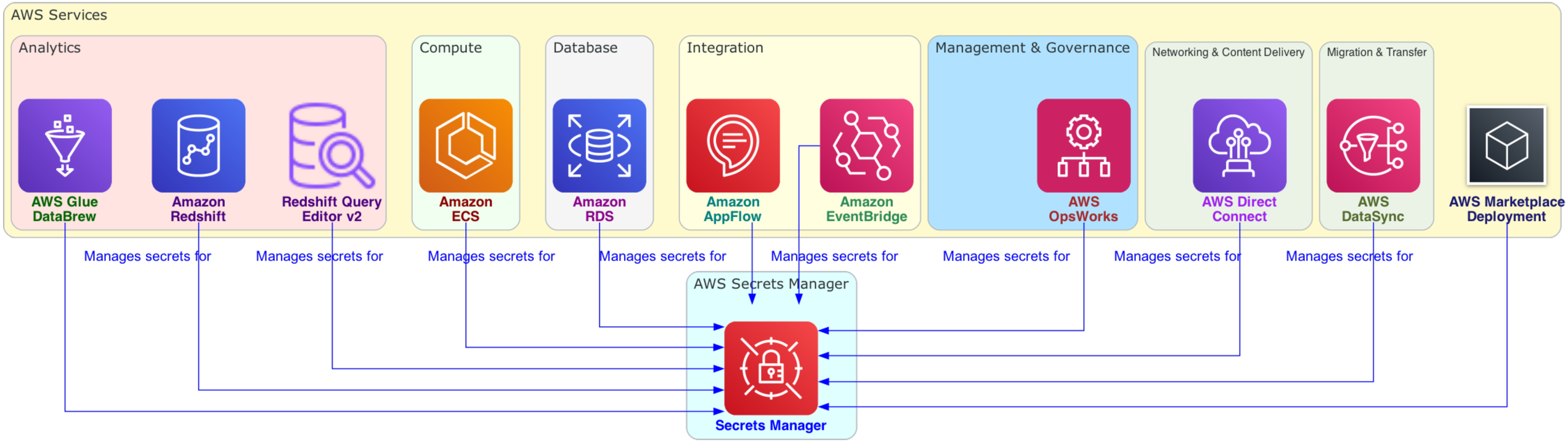
🛠️ Managed rotation

🔐 For most managed secrets

⚙️ Service configures and manages rotation

🚫 Doesn't use Lambda function

AWS Secrets Manager secrets managed by other AWS services



1. 🗝️ Many AWS services store, use secrets
🔐 Secrets Manager

2. 🛡️ Some secrets are managed secrets
🔧 Service helps manage them
🔄 Managed rotation included
🚫 Restrictions on updating, deleting
⚠️ Prevents outages

3. 📛 Managed secrets naming convention
🆔 Includes managing service ID
🔑 Secret name format: ServiceID!MySecret
🪪 `Secret ARN format: arn:aws:us-east-1:ServiceID!MySecret-a1b2c3`

```
appflow, databrew, datasync, directconnect, ecs-sc, events, marketplace-deployment, opsworks-cm, rds, redshift, sqlworkbench
```

Thanks for Watching