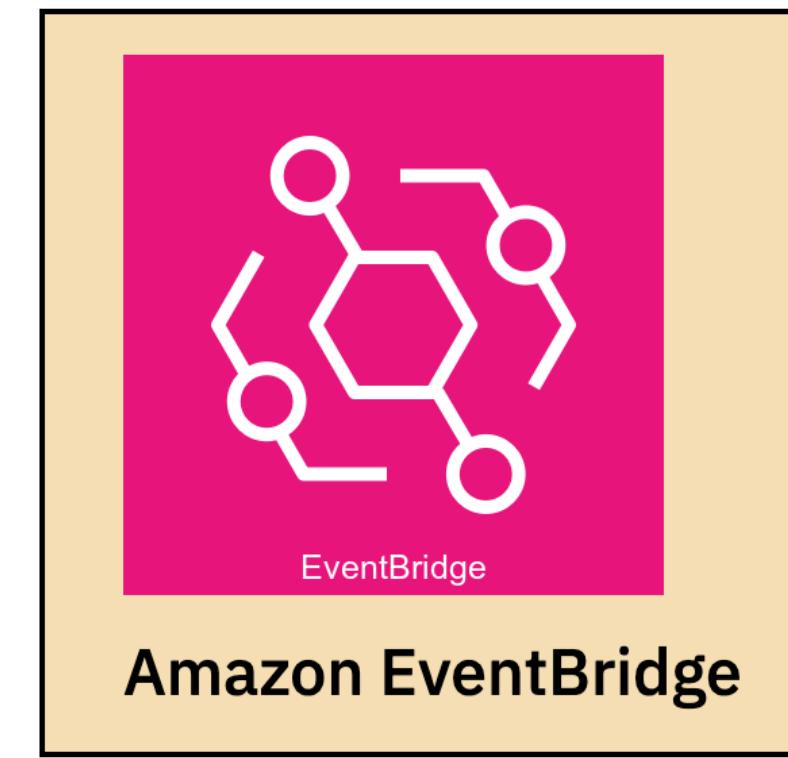




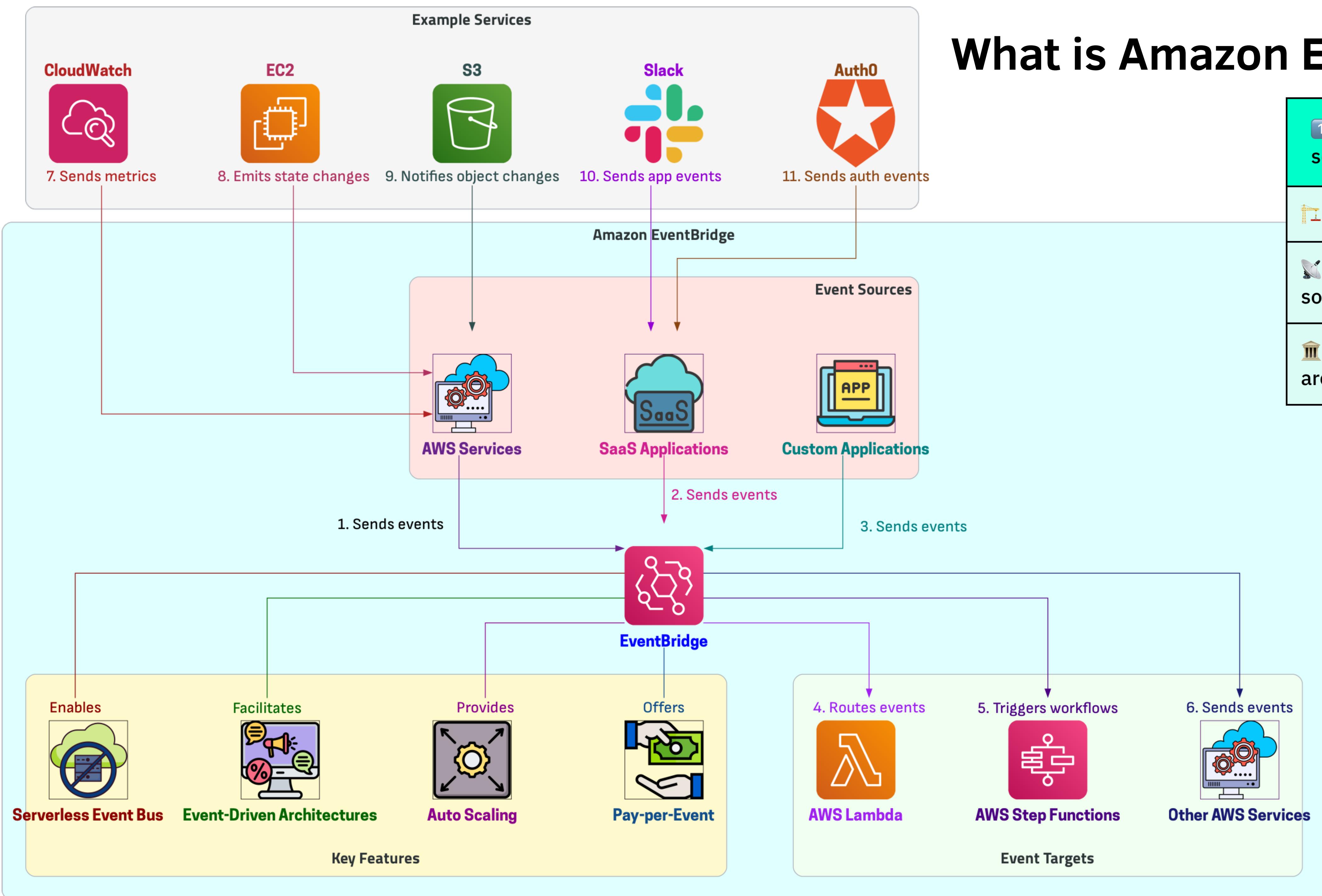
# Amazon EventBridge

# Table of Contents

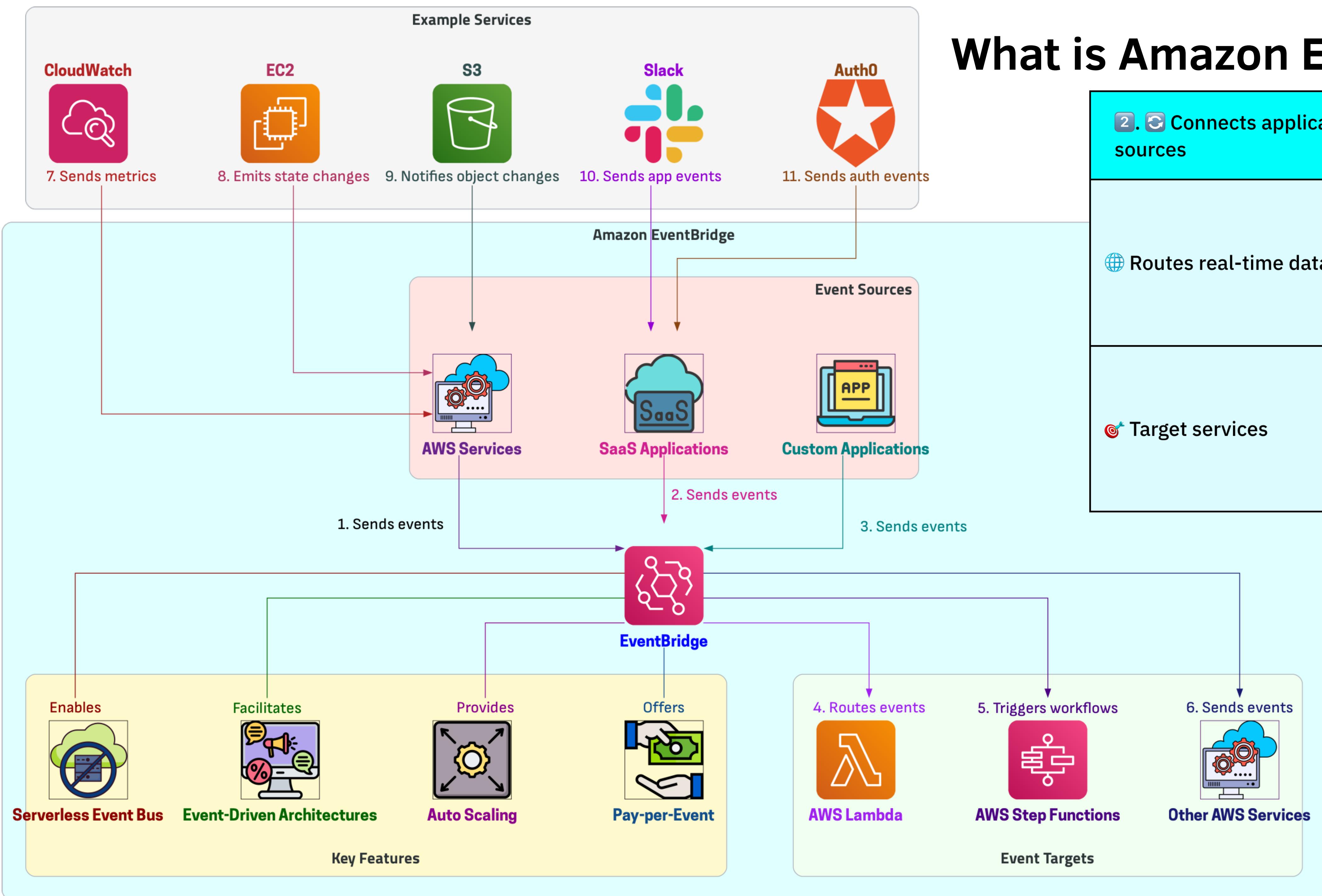


- 1. What is Amazon EventBridge?
- 2. High Level Overview
- 3. EventBridge: The Evolution of Amazon CloudWatch Events
- 4. Amazon EventBridge Events
- 5. Amazon EventBridge Event Patterns
- 6. Amazon EventBridge Rules
- 7. Amazon EventBridge Targets
- 8. EventBridge Pipes
- 9. EventBridge Pipes: Key Concepts
- 10. Amazon EventBridge: Global Endpoints & Event Replication
- 11. Amazon EventBridge Schemas

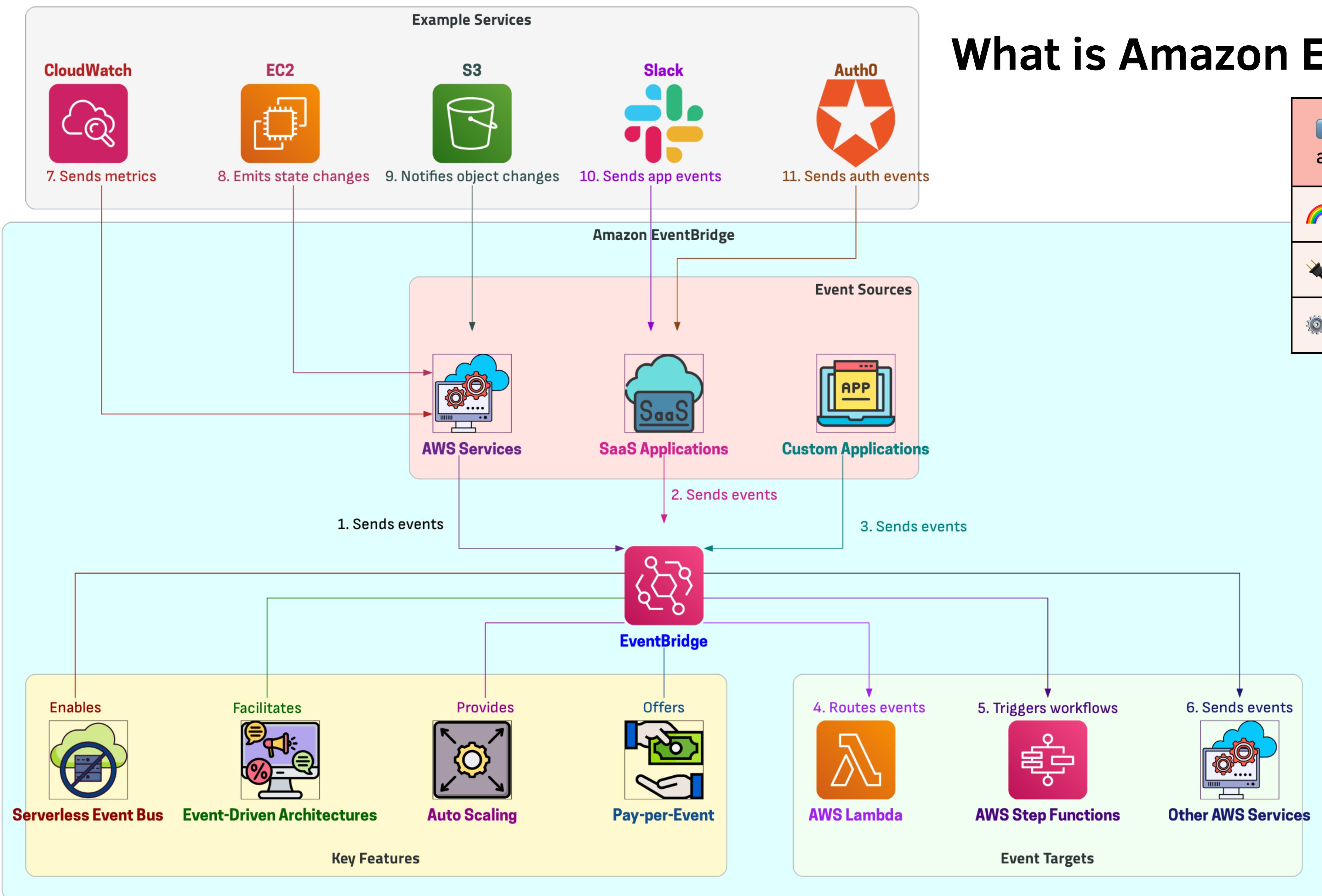
# What is Amazon EventBridge?



# What is Amazon EventBridge?

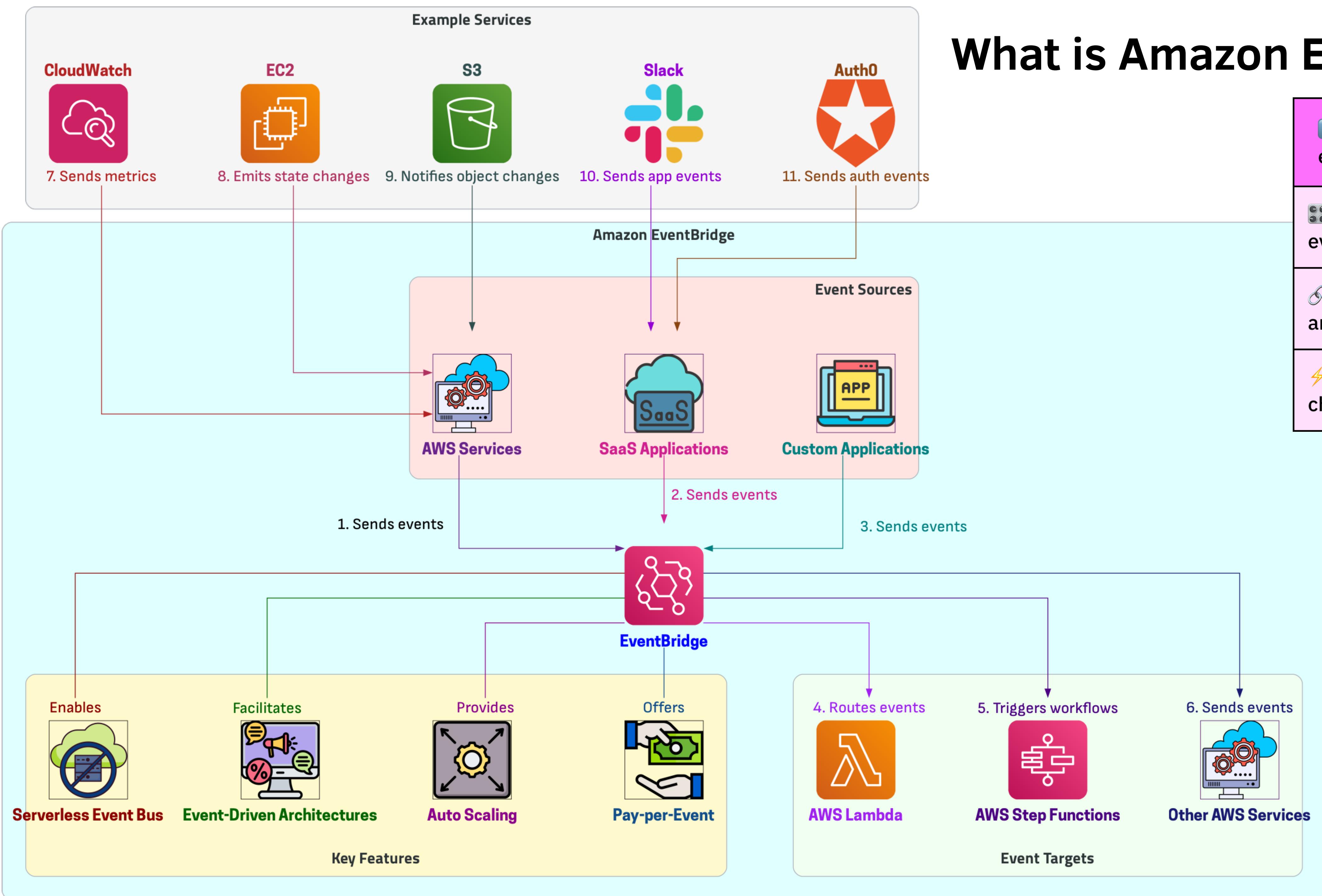


# What is Amazon EventBridge?



- 3. Supports 100+ AWS and SaaS event sources
- Wide range of sources
- Simple integration
- Easy event processing

# What is Amazon EventBridge?



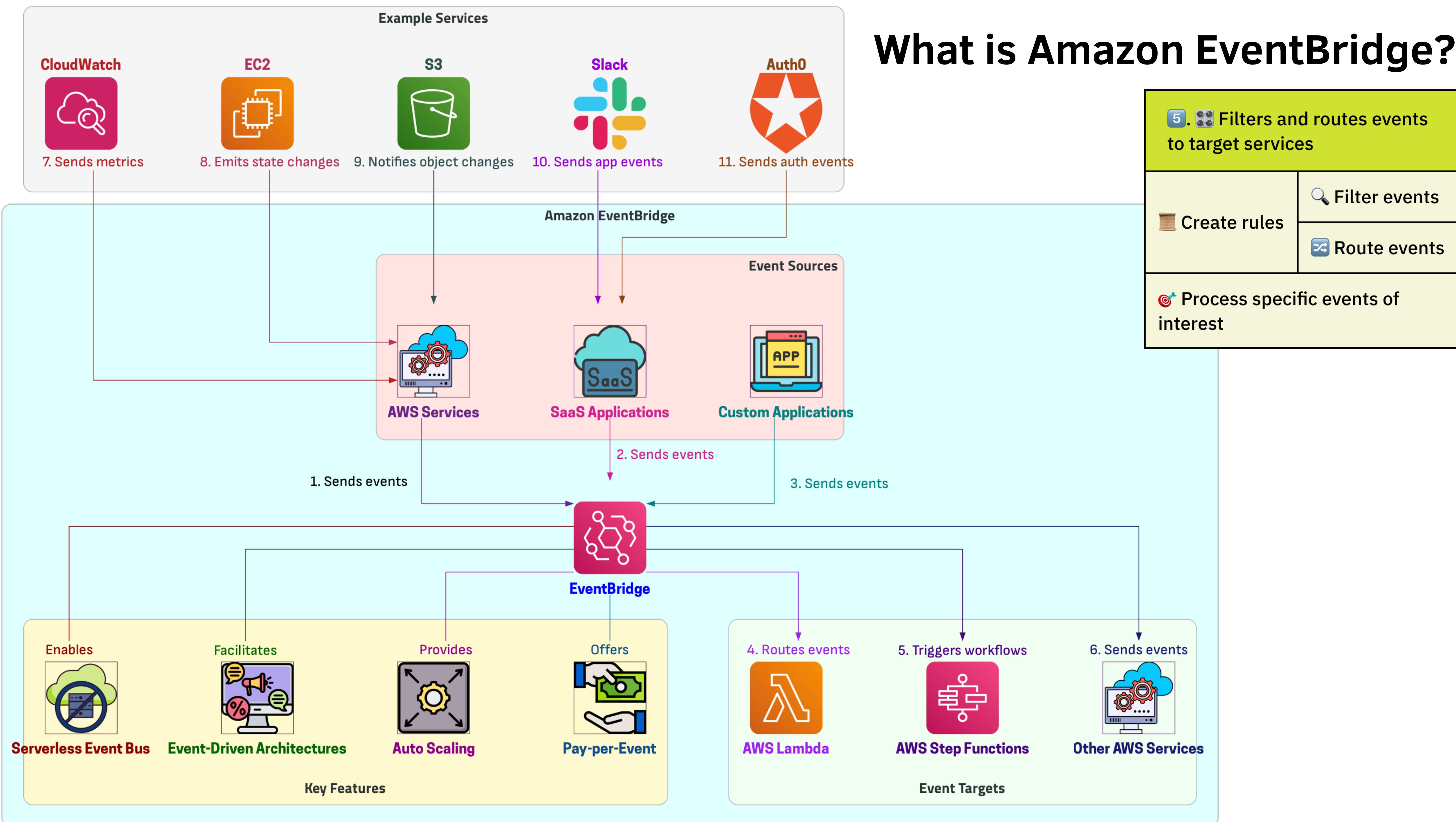
4. Enables building event-driven architectures

Central platform for event management

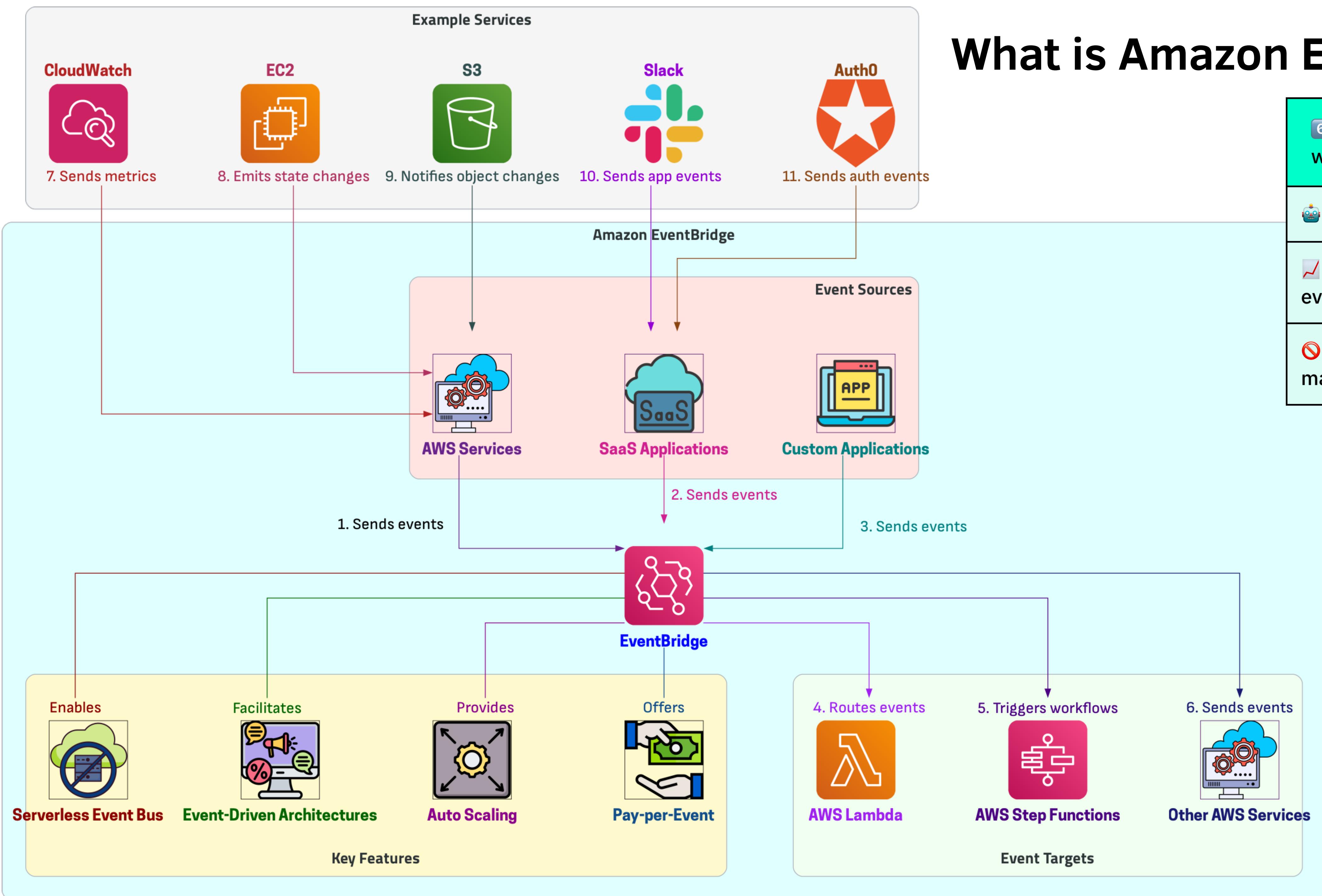
Loosely coupled architectures

Real-time reactions to changes

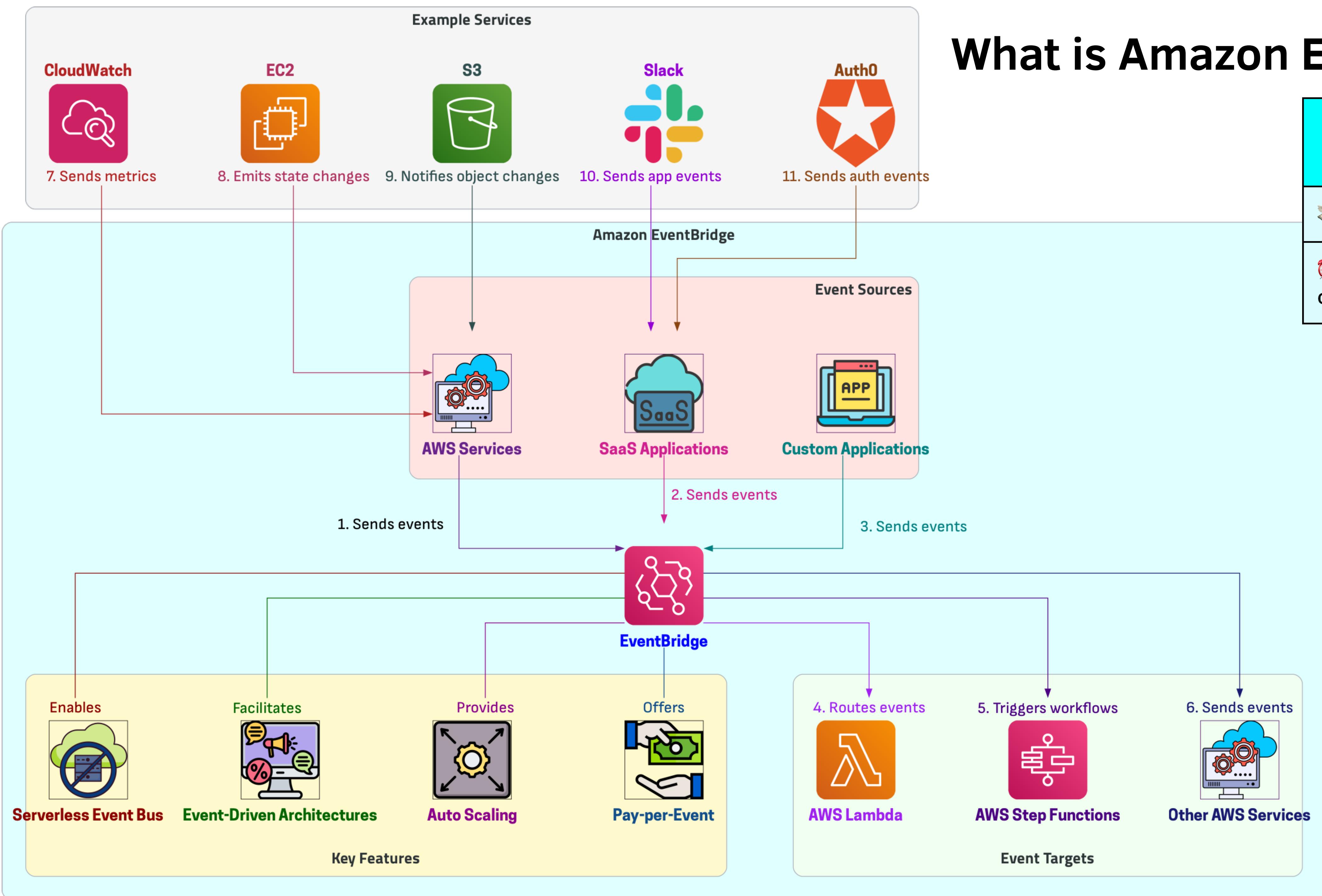
# What is Amazon EventBridge?



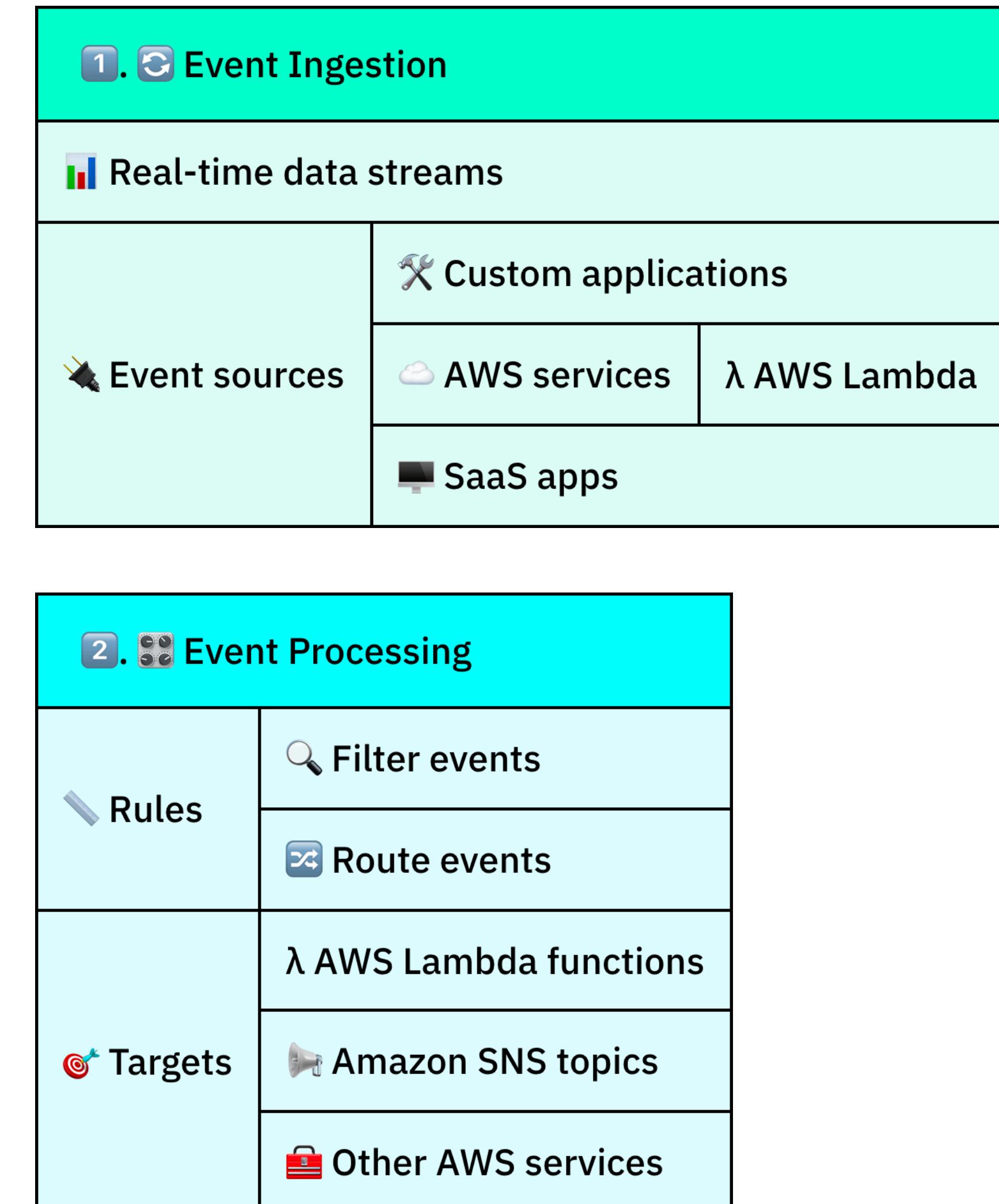
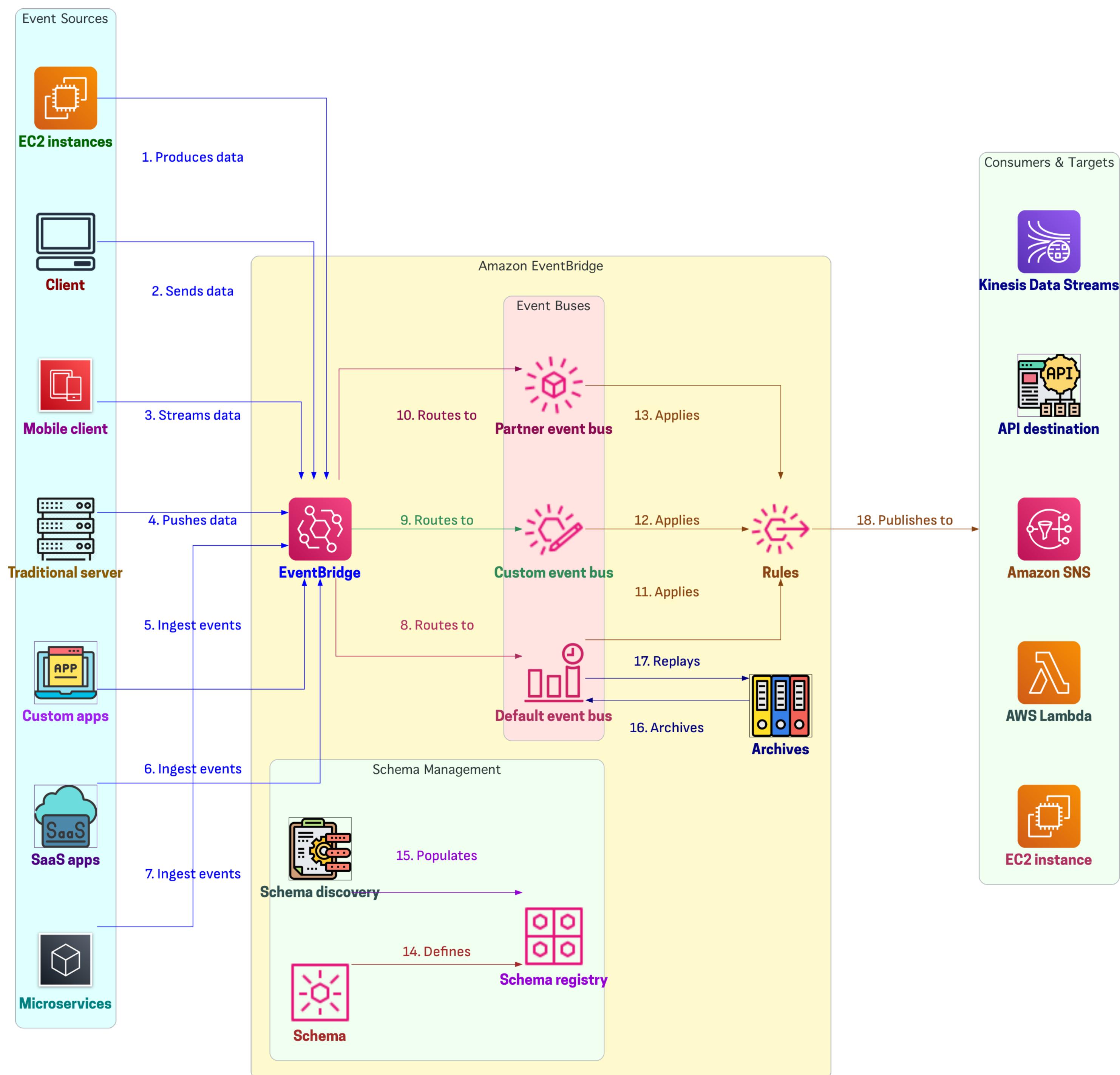
# What is Amazon EventBridge?



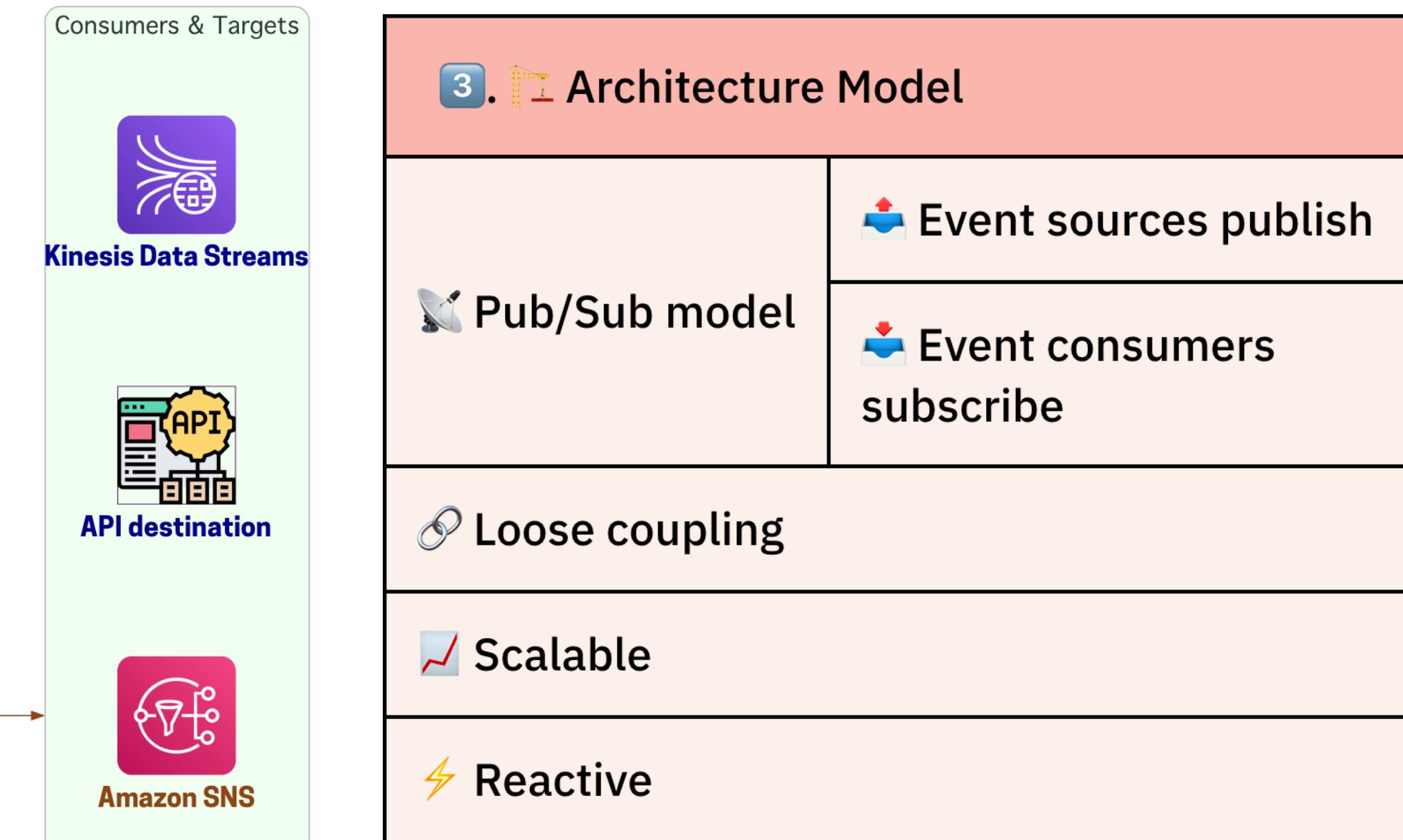
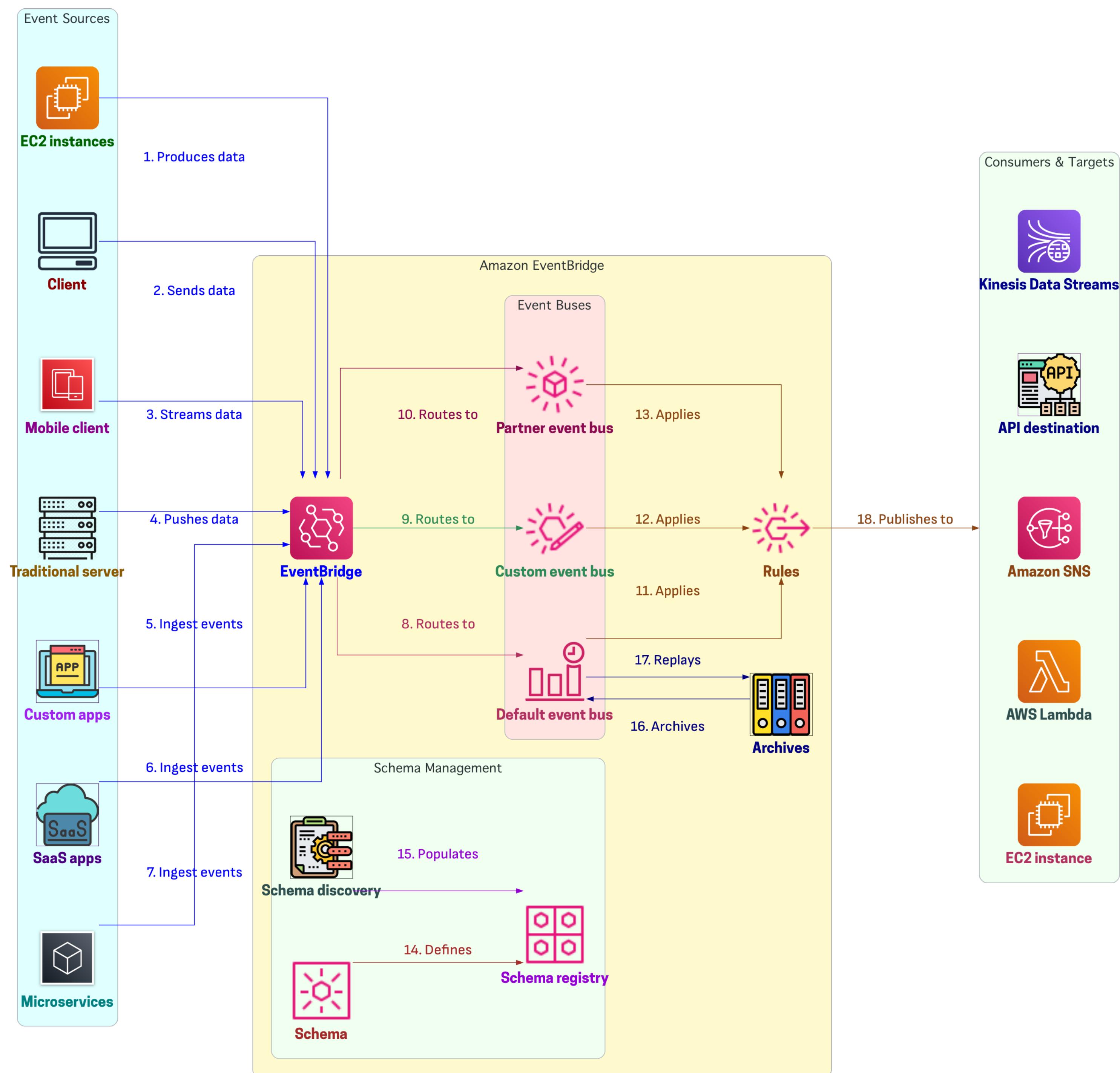
# What is Amazon EventBridge?



# High Level Overview

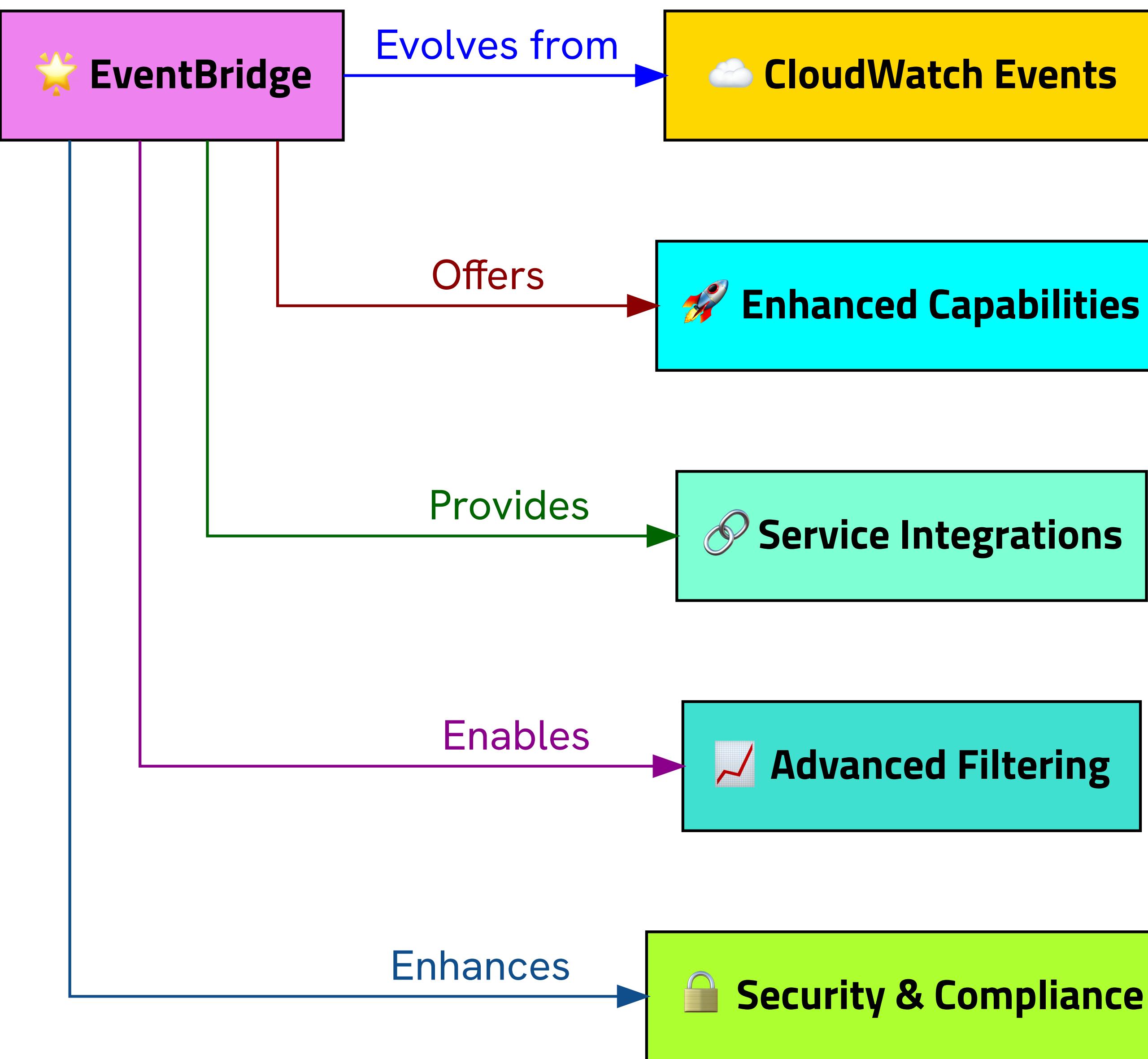


# High Level Overview

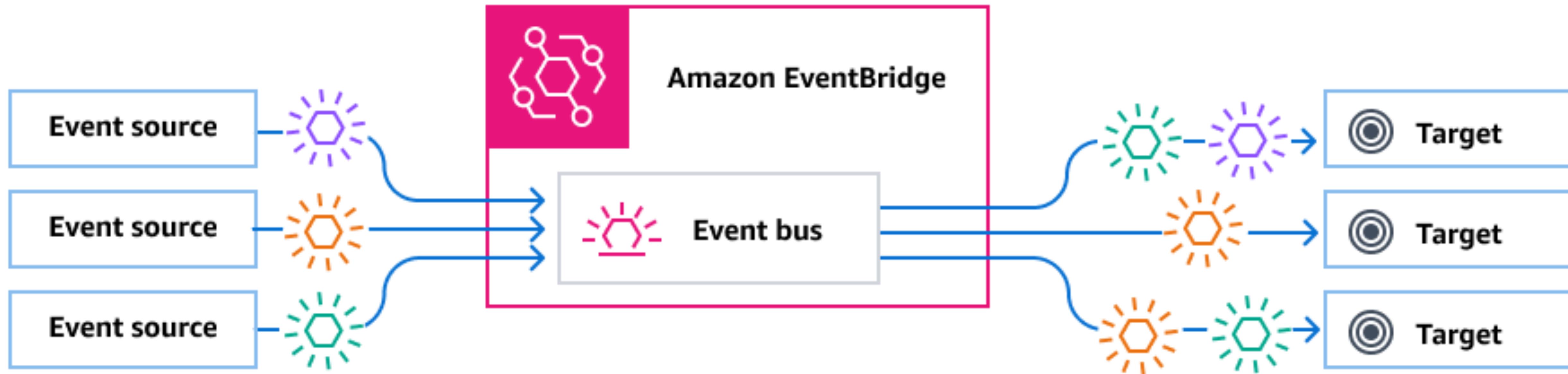




# EventBridge: The Evolution of Amazon CloudWatch Events



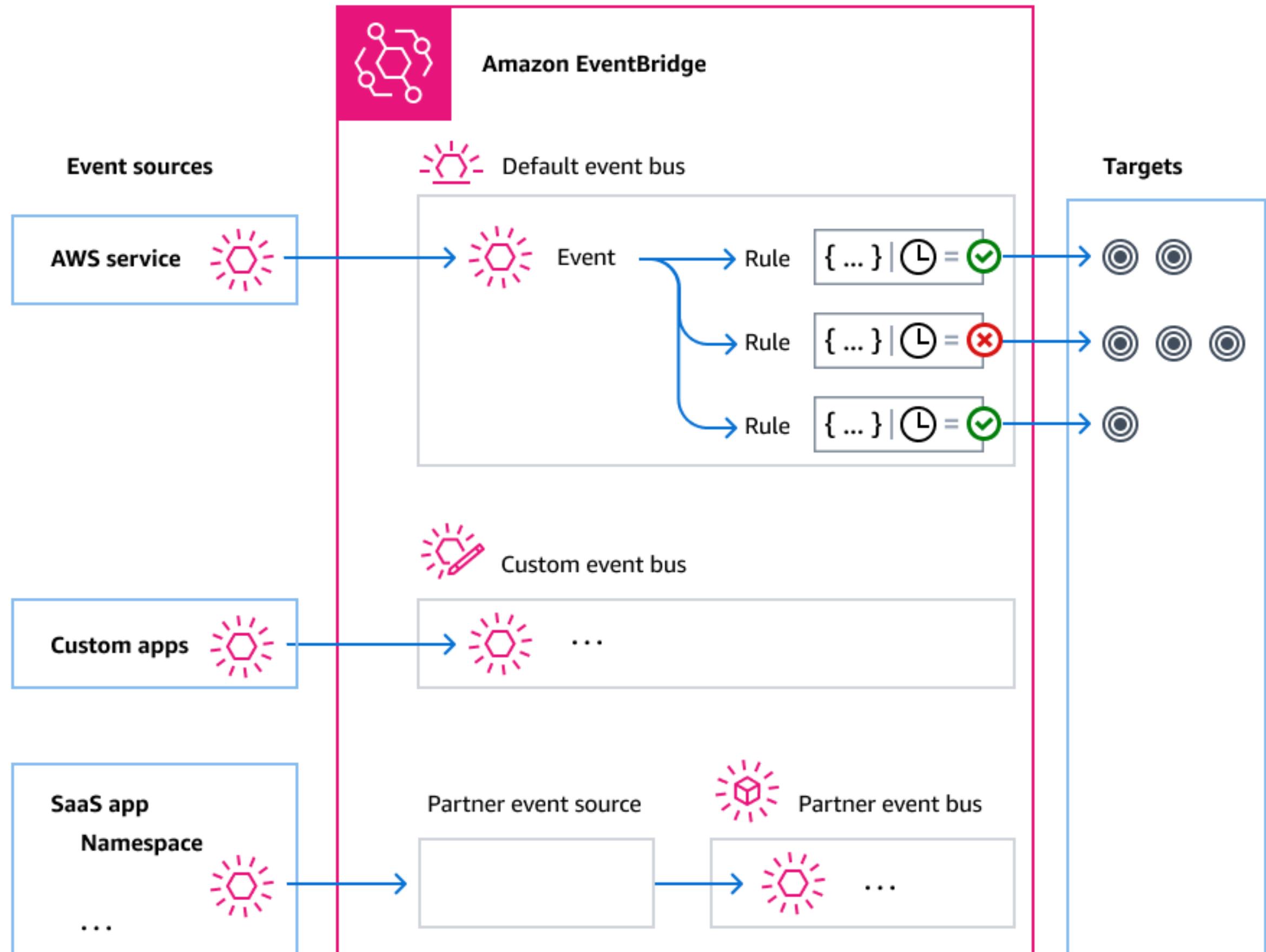
1. From CloudWatch Events to EventBridge
  - 🚀 Significant leap from predecessor
  - 🚏 More flexible, powerful event bus service
2. Enhanced event handling capabilities
  - 🚀 Advanced features
  - 🏗️ Enables complex application architectures
3. Broader service integrations
  - ☁️ AWS services
  - 💻 SaaS applications
  - 🔧 Custom applications
  - 🌉 Easier connectivity across environments
4. Advanced event filtering options
  - 🎯 Precise targeting
  - 🔍 Improved filtering mechanisms
  - ⚙️ Processing based on specific criteria
5. Safer, more secure event management
  - 🔒 Enhanced security
  - 📜 Improved compliance
  - 🤝 Reliable, secure communication between services



## 💡 Amazon EventBridge Event Bus

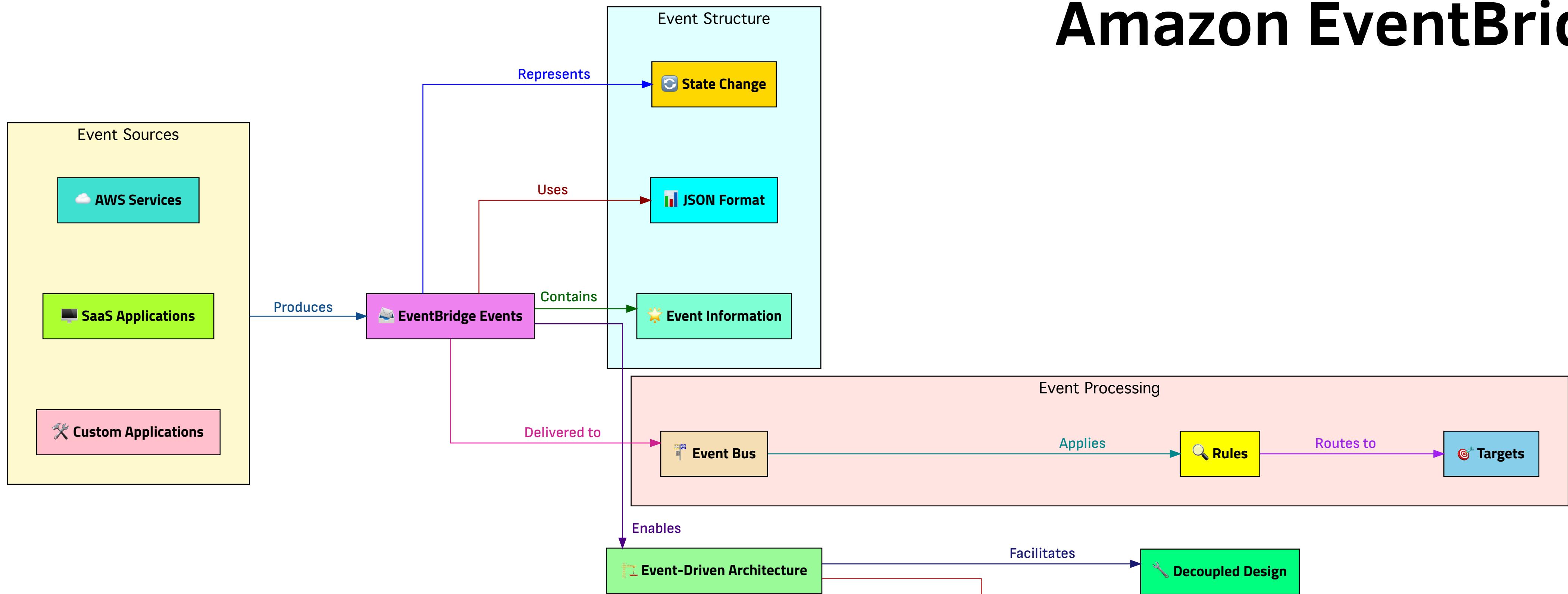
- 🎯 **Central hub for receiving and routing events:** 🏛️ Receives events from various sources, ⚡ Routes events to target services based on rules
- 🌐 **Event bus for each AWS account and region:** 🌎 Default event bus per account and region, 🚀 Easy to get started with event-driven architectures
- 🔄 **Custom event buses for specific applications:** 🎨 Create custom event buses for specific use cases, 🔍 Isolate events and maintain separate event streams
- 🔌 **Connects event sources and target services:** 🌈 Event sources: AWS services, SaaS, custom apps, 🎯 Target services: Lambda, SNS, SQS, etc.
- 🧩 **Enables decoupled, event-driven architectures:** 🔗 Event producers and consumers independent, 📈 Promotes flexibility and scalability
- 📡 **Supports AWS and partner event sources:** 💫 Wide range of AWS services supported, 🤝 Simple integration and routing of events
- 🔒 **Configurable permissions for event buses:** 🔒 Configure permissions using IAM, 🗂️ Secure and granular control over pub/sub

## 💡 How Amazon EventBridge Event Buses Work



1. 📧 **Receive events from various sources:** 🌐 AWS services, 🖱️ SaaS applications and custom applications
2. 🔎 **Apply rules to filter and route events:** 🔎 Match events based on content and structure, ⚡ Filter and route to specific targets
3. 🎯 **Send matched events to specified targets:** 🎯 AWS Lambda, SNS topics, SQS queues, 🚀 Other AWS services
4. 🔍 **Enable event-driven communication:** 🔍 Between decoupled components, 🔍 Real-time reactivity without direct dependencies
5. 🔗 **Decouple event producers and consumers:** 🔗 Flexibility, scalability, maintainability, 🔧 Architectural improvement
6. 📈 **Scale automatically to handle event load:** 🚀 High volume handling, 📊 No manual management
7. 🔒 **Control access with IAM permissions:** 🔒 Authorized producers/senders, 🔒 Authorized receivers/consumers

# Amazon EventBridge Events



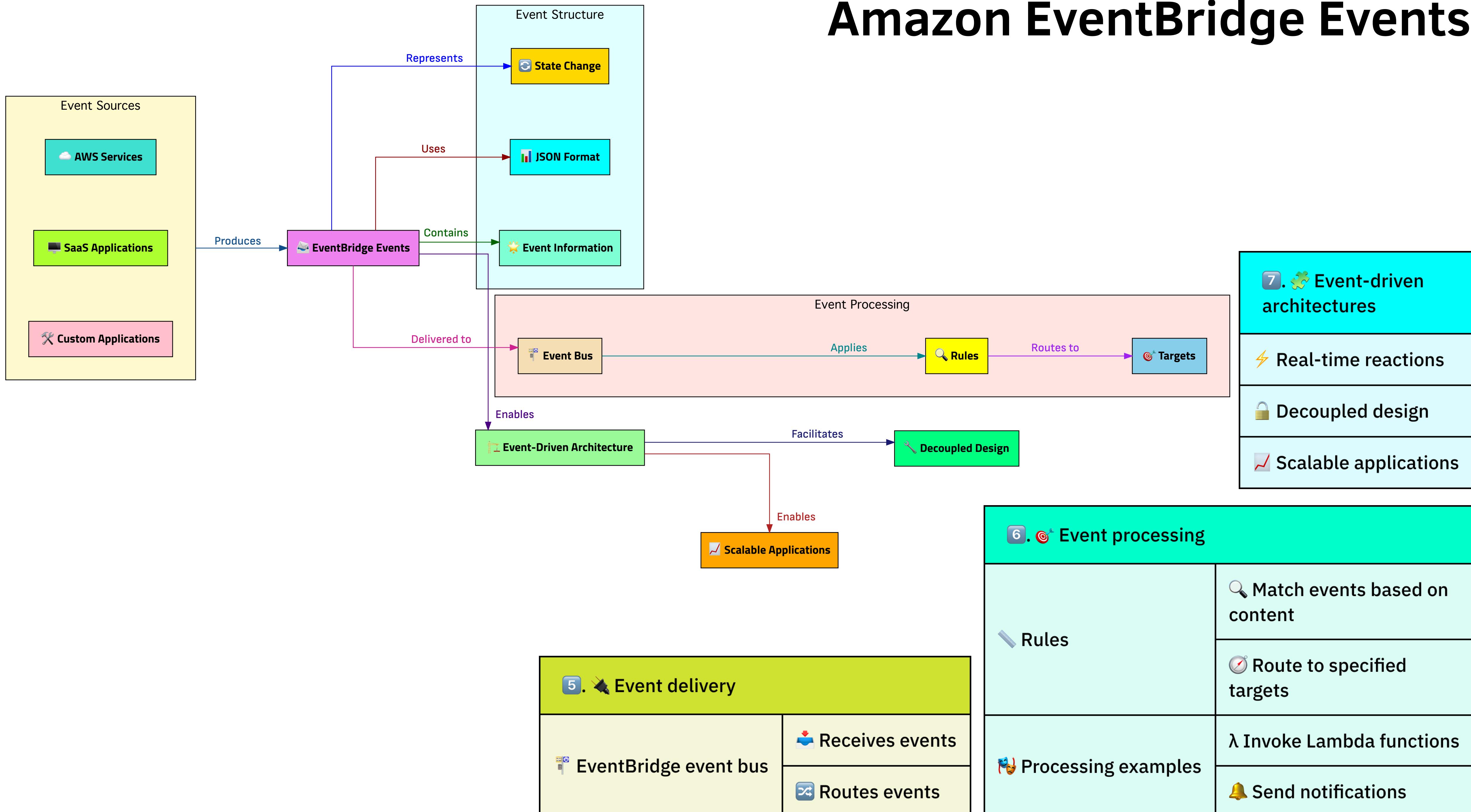
1. 📱 Events as changes in state or updates
Change in AWS resource
New file uploaded to S3
Commit pushed to GitHub

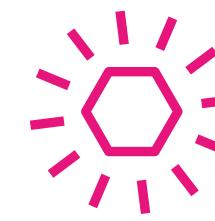
2. 📦 JSON-formatted data structures
Standardized information exchange
Between event producers and consumers

3. 🔎 Event components
Event source
Detail-type
Specific event details

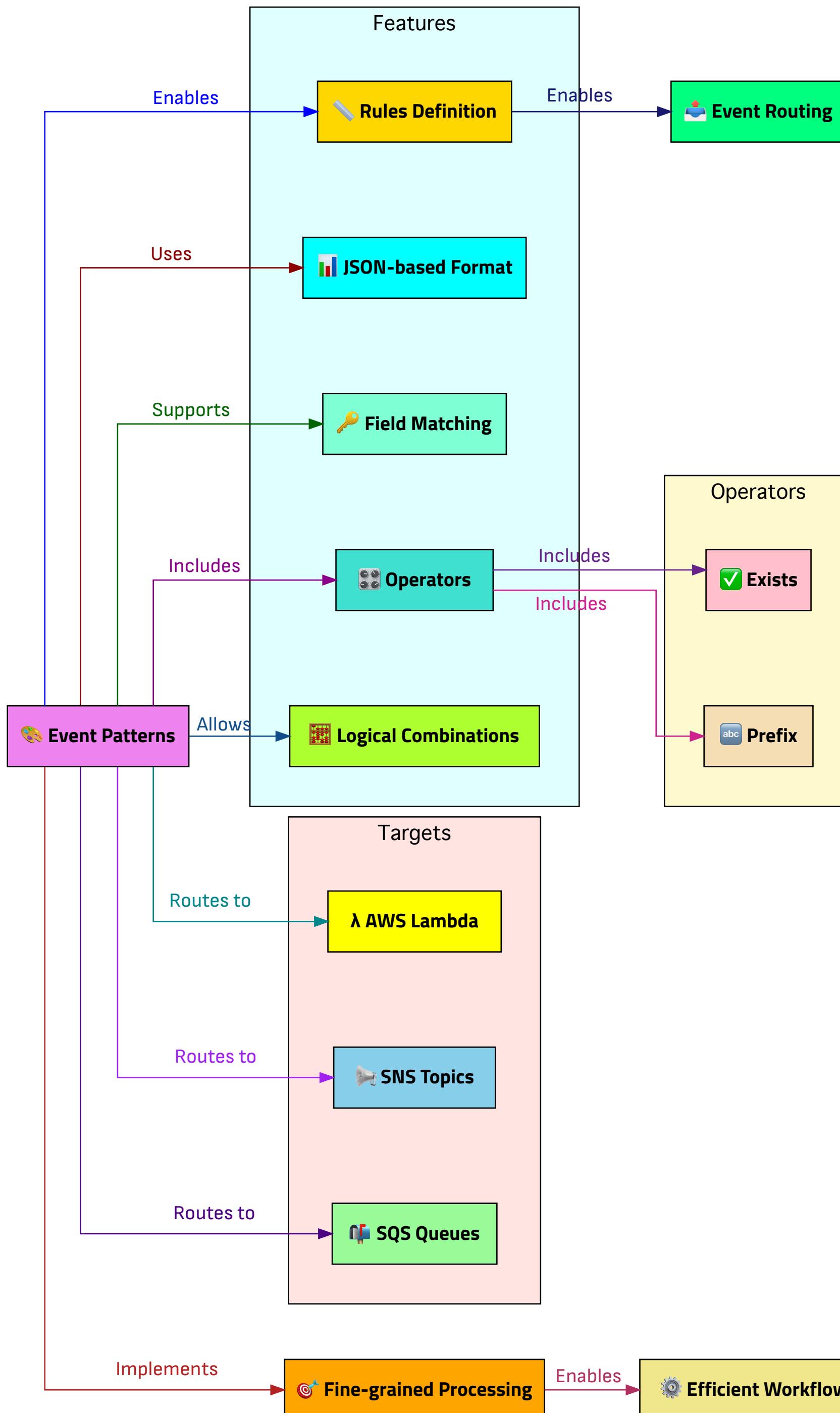
4. 🌐 Event sources	
AWS services	EC2 S3
SaaS applications	Zendesk Datadog
Custom applications	Using EventBridge API

# Amazon EventBridge Events





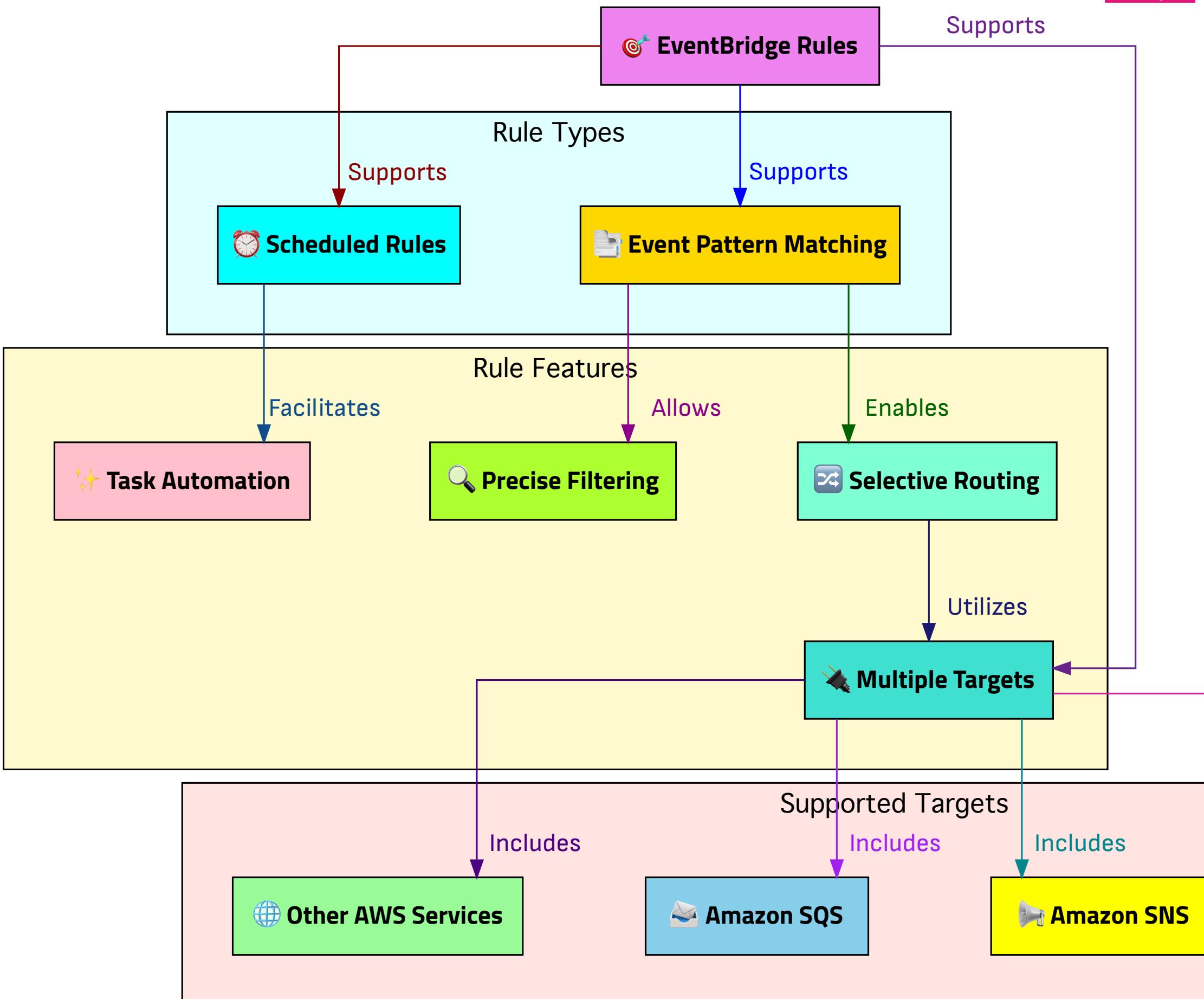
# Amazon EventBridge Event Patterns



1. Define rules to match events
  - Selective event processing
  - Event routing
2. JSON-based format for filtering
  - Specify matching criteria
  - Complex filtering logic
3. Match on event fields and values
  - Event source
  - Detail-type
  - Custom fields in event detail
4. Use operators


'exists' operator	Check field presence
'prefix' operator	Match based on field value prefix
5. Combine multiple conditions
  - Logical AND
  - Logical OR
  - Sophisticated matching rules
6. Route matched events to targets
  - AWS Lambda functions
  - SNS topics
  - SQS queues
7. Enable fine-grained event processing
  - Process relevant events
  - Efficient workflows
  - Targeted event-driven architecture

# Amazon EventBridge Rules



**5. Parallel processing**

- ⚡ Optimized response times
- 📊 Efficient resource utilization

**6. Customizable rules**

- ⚙️ Flexible event matching
- 🔍 Precise filtering conditions

- 1. Two types of rules**
  - ⌚ Event pattern matching
  - 🕒 Scheduled
- 2. Event pattern matching**
  - 🔍 Define specific data criteria
  - ⌚ Match specific data criteria
- 3. Scheduled rules**
  - 🤖 Automate routine tasks
  - 📅 Trigger at specified intervals
- 4. Multiple target routing**
  - λ AWS Lambda
  - SNS
  - SQS
  - Other AWS services
- 5. Parallel processing**
  - ⚡ Optimized response times
  - 📊 Efficient resource utilization
- 6. Customizable rules**
  - ⚙️ Flexible event matching
  - 🔍 Precise filtering conditions
- 7. Task automation**
  - 📅 Scheduled triggers
  - ✨ Eliminate manual intervention

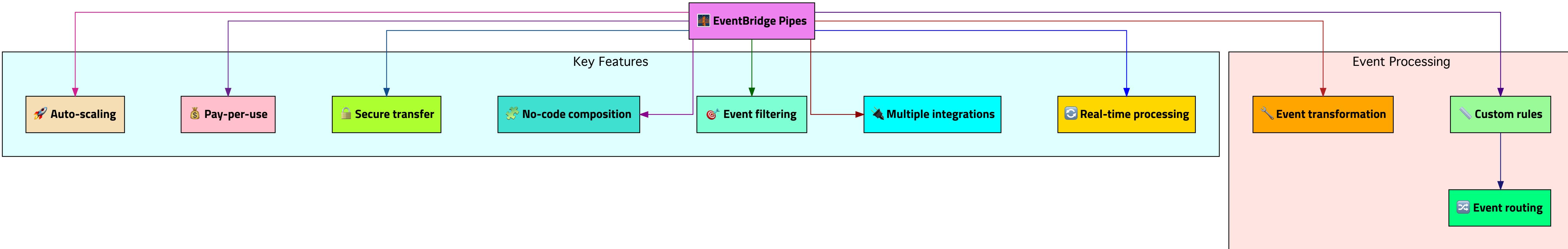


# Amazon EventBridge Targets

1. API Destination
2. API Gateway
3. AWS AppSync
4. Batch Job Queue
5. CloudWatch Log Group
6. CodeBuild Project
7. CodePipeline
8. EBS CreateSnapshot API Call
9. EC2 Related API Calls
10. ECS Task
11. Event Bus in Different Account/Region
12. Firehose Delivery Stream
13. Glue Workflow
14. Incident Manager Response Plan
15. Inspector Assessment Template
16. Kinesis Stream
17. Lambda Function (ASYNC)
18. Amazon Redshift Data API Queries
19. SageMaker Pipeline
20. Amazon SNS Topic (Non-FIFO)
21. Amazon SQS Queue
22. Step Functions State Machine (ASYNC)

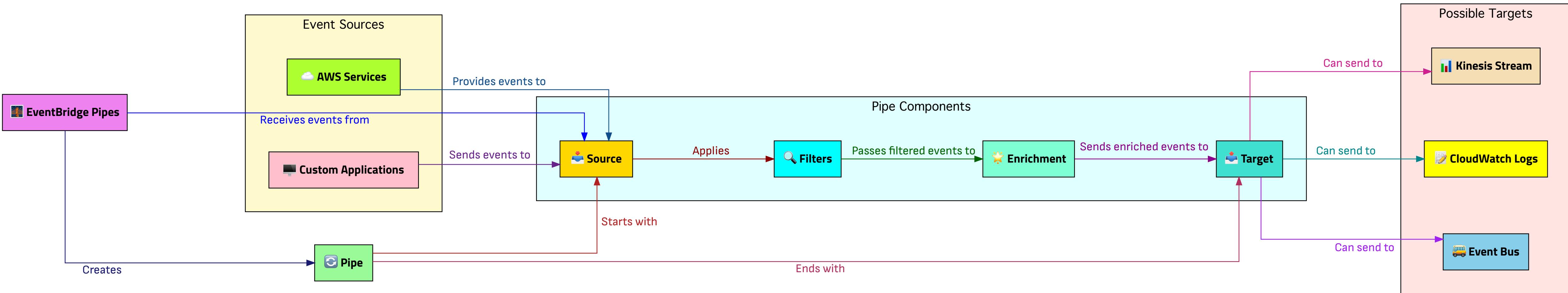


# EventBridge Pipes



1. **Real-time event processing:** 🛫 Events from various sources, ⏳ Timely data flow, ⚡ Responsiveness
2. **100+ AWS and SaaS integrations:** 🧩 Integrates with AWS services, 🌐 Third-party SaaS applications, 🌈 Diverse event sources
3. **Filters events based on rules:** 🔎 Custom filtering rules, 🚦 Event routing based on criteria, 🎯 Processes relevant events only
4. **No-code event composition:** 🖍️ No-code interface, 🔄 Composing, transforming events, 🎭 Event-driven workflows, ❌ No complex coding required
5. **Secure data transfer:** 🔒 Ensures data confidentiality, 🛡️ Maintains data integrity, 🛡️ Between event sources and targets
6. **Pay-per-use pricing model:** 💰 Pay only for processed events, 💲 Cost-effective solution, 🚧 Event-driven architectures
7. **Scales automatically:** 🚀 Handles varying event volumes, ⚖️ Ensures reliable performance, 🏆 During peak traffic periods

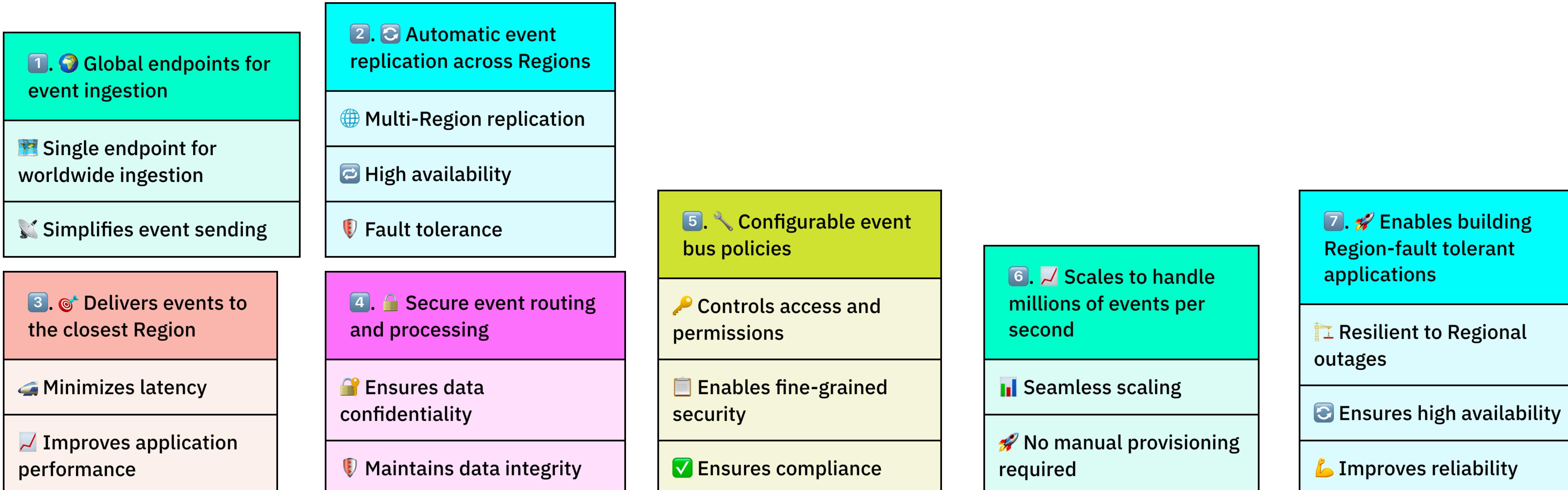
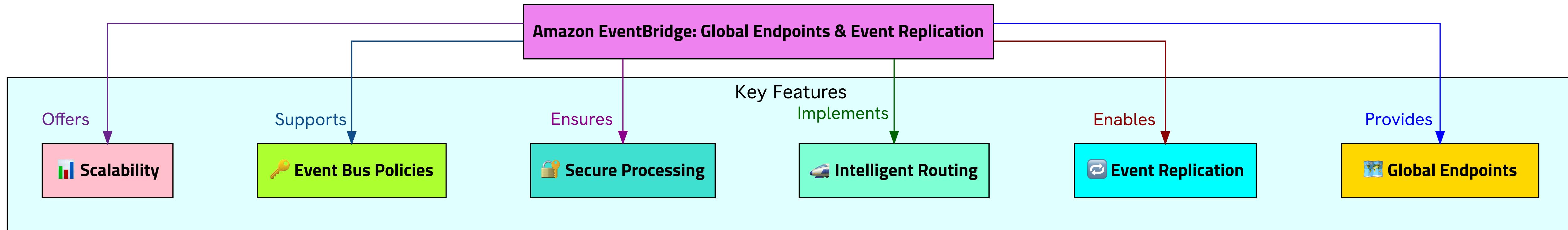
# EventBridge Pipes: Key Concepts



1. **⟳ Pipe:** Routes events from source to target: **⟳** Single source to single target, **🔍** Filters specific events, **⭐** Performs enrichments on event data
2. **📥 Source:** Receives events from various services: **📥** Receives event data, **🔍** Applies filters and enrichment, **📤** Sends to target, **🔢** Maintains event order
3. **🔍 Filters:** Processes a subset of events based on patterns: **🔍** Filters source's events, **🔢** Processes subset of events, **📝** Defines event pattern for filtering, **💰** Charged only for matched events
4. **⭐ Enrichment:** Enhances event data before sending to target: **⭐** Enhances event data, **📝** Example: Ticket created events, **🔍** Uses Lambda to get full ticket details, **📤** Sends enriched data to target
5. **📤 Target:** Destination for filtered and enriched events: **📤** Receives filtered and enriched events, **🌊** Example targets: Kinesis, CloudWatch, **🔄** Allows data transformation before sending
6. **🎯 Multiple pipes can send events to the same target:** **🎯** Multiple pipes, **📥** Different sources, **📤** Same target
7. **🚌 Pipes and event buses can work together for multiple targets:** **🚌** Pipes and event buses together, **🔄** Pipe sends events to event bus, **🚌** Event bus sends events to multiple targets, **📝** According to specified rules

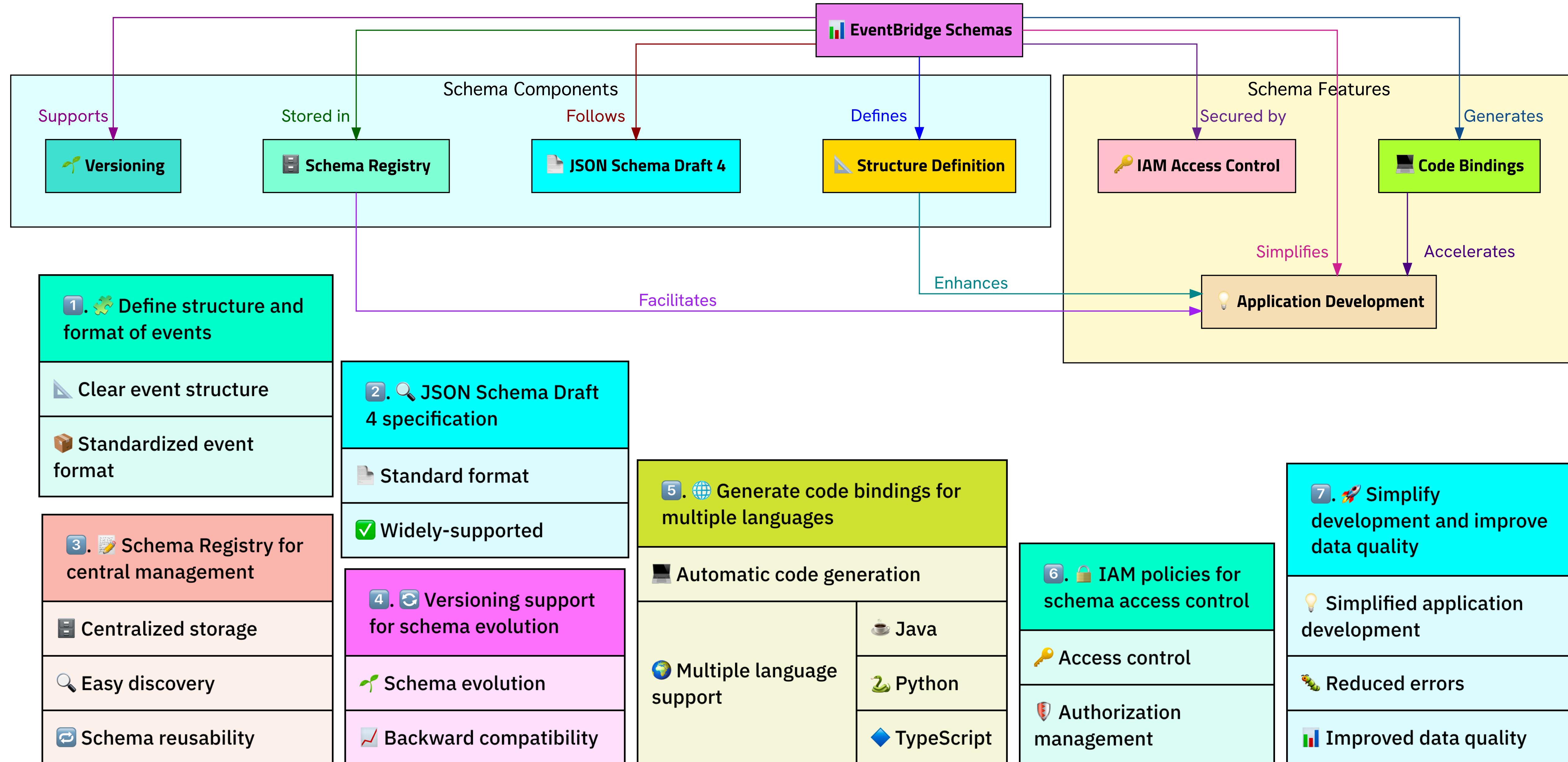


# Amazon EventBridge: Global Endpoints & Event Replication





# Amazon EventBridge Schemas





**Thanks  
for  
Watching**