

# Event-Driven Architecture Decision Reference (Version 1.0)

**Συγγραφέας:** Enterprise Architecture Group

**Σκοπός:** Παροχή οδηγιών για την επιλογή του κατάλληλου event-driven pattern σε διαφορετικά επιχειρησιακά και τεχνικά σενάρια.

## 1. Εισαγωγή

Το Event-Driven Architecture (EDA) είναι ένα κατανεμημένο αρχιτεκτονικό στυλ όπου η επικοινωνία γίνεται μέσω events, μηνυμάτων ή streams αντί για άμεσες συγχρονισμένες κλήσεις. Ο οδηγός αυτός ορίζει πότε να χρησιμοποιείτε κάθε EDA pattern και πώς να αξιολογείτε την αρχιτεκτονική καταλληλότητα.

## 2. Τύποι Events

### 2.1 Notification Events

Σήματα fire-and-forget που ενημερώνουν ότι συνέβη κάτι. Χρήση για ελαφριές ενημερώσεις.

### 2.2 State Transfer Events

Περιέχουν την πλήρη κατάσταση μιας οντότητας μετά από μια αλλαγή. Χρήση όταν οι downstream consumers χρειάζονται πλήρες context.

### 2.3 Event-Carried State Transfer

Τα events κωδικοποιούν δεδομένα domain που επιτρέπουν στους consumers να ενημερώσουν τα δικά τους εσωτερικά μοντέλα. Υποστηρίζει ανεξάρτητα read models και CQRS.

## 3. EDA Integration Patterns

### 3.1 Event Streaming

**Χρήση όταν:**

- Η ταχύτητα δεδομένων είναι υψηλή ή πολύ υψηλή.
- Οι απαιτήσεις latency είναι σχεδόν πραγματικού χρόνου.
- Πολλοί consumers χρειάζονται το ίδιο event stream.
- Analytics, real-time dashboards, anomaly detection.

#### **Αποφύγετε όταν:**

- Απαιτείται αυστηρή συνέπεια.
- Οι consumers εξαρτώνται από συγχρονισμένες απαντήσεις.
- Η σειρά των events είναι κρίσιμη (εκτός αν οι τεχνολογίες το υποστηρίζουν).

#### **Τυπικές τεχνολογίες:**

- Kafka
- Azure Event Hubs
- Confluent Platform

## **3.2 Event-Driven Orchestration**

#### **Χρήση όταν:**

- Οι επιχειρησιακές διαδικασίες απαιτούν ελεγχόμενη ακολουθία.
- Υπάρχουν βήματα ανθρώπινης έγκρισης.
- Χρειάζεστε robust retry, compensation ή monitoring.
- Το latency είναι δευτερόλεπτα έως λεπτά.

#### **Αποφύγετε όταν:**

- Απαιτείται ακραία κλιμάκωση ή υψηλός όγκος events.
- Το workflow μπορεί να χορογραφηθεί πλήρως χωρίς κεντρικό έλεγχο.

#### **Τυπικές τεχνολογίες:**

- Azure Durable Functions
- Logic Apps
- AWS Step Functions

## **3.3 Event-Driven Choreography**

#### **Χρήση όταν:**

- Οι υπηρεσίες είναι ανεξάρτητες και loosely coupled.
- Η διαδικασία μπορεί να εξελιχθεί χωρίς κεντρικό συντονισμό.
- Απαιτείται κλιμάκωση και ανθεκτικότητα.

- Χρησιμοποιείται domain-driven design και bounded contexts.

#### **Αποφύγετε όταν:**

- Απαιτείται end-to-end visibility.
- Η σύνθετη επιχειρησιακή λογική εκτείνεται σε πολλές υπηρεσίες.
- Χρειάζεστε transactional integrity μεταξύ domains.

#### **Τυπικές τεχνολογίες:**

- Service Bus Topics
- Event Grid
- Kafka

### **3.4 Command/Event Hybrid (CQRS + Event Sourcing)**

#### **Χρήση όταν:**

- Απαιτείται ιστορικό audit trail.
- Χρειάζεται ανακατασκευή της κατάστασης του συστήματος από events.
- Οι εγγραφές απαιτούν αυστηρή domain validation.
- Τα read models διαφέρουν από τα write models.

#### **Αποφύγετε όταν:**

- Το domain είναι απλό και αρκεί το CRUD.
- Η διαχείριση event inflation ή versioning γίνεται μη διαχειρίσιμη.
- Δεν απαιτείται replayable ιστορικό.

#### **Τυπικές τεχνολογίες:**

- EventStoreDB
- Kafka με compacted topics
- Azure Cosmos DB (event sourcing style)

## **4. Decision Signals for EDA Domain**

### **Functional Signals**

- event\_volume: low | medium | high | very\_high
- event\_latency: minutes | seconds | near\_real\_time
- process\_control\_style: orchestration | choreography | hybrid
- auditability\_requirement: low | medium | strict

- data\_consistency: eventual | strong
- event\_type: notification | state\_transfer | event\_sourcing

## Non-Functional Signals

- scalability\_needs: low | medium | high
- fault\_tolerance: basic | advanced
- operational\_complexity: low | medium | high
- consumer\_count: single | multiple | dynamic

## 5. Summary Table

<b>Pattern</b>	<b>Best When</b>	<b>Avoid If</b>	<b>Latency</b>	<b>Event Volume</b>
Event Streaming	high velocity	strict consistency	ms-sec	high--very high
Orchestration	controlled flow	high throughput	sec-min	medium
Choreography	independent services	visibility required	sec	high
CQRS + Event Sourcing	audit + replay	simple domain	sec-min	medium