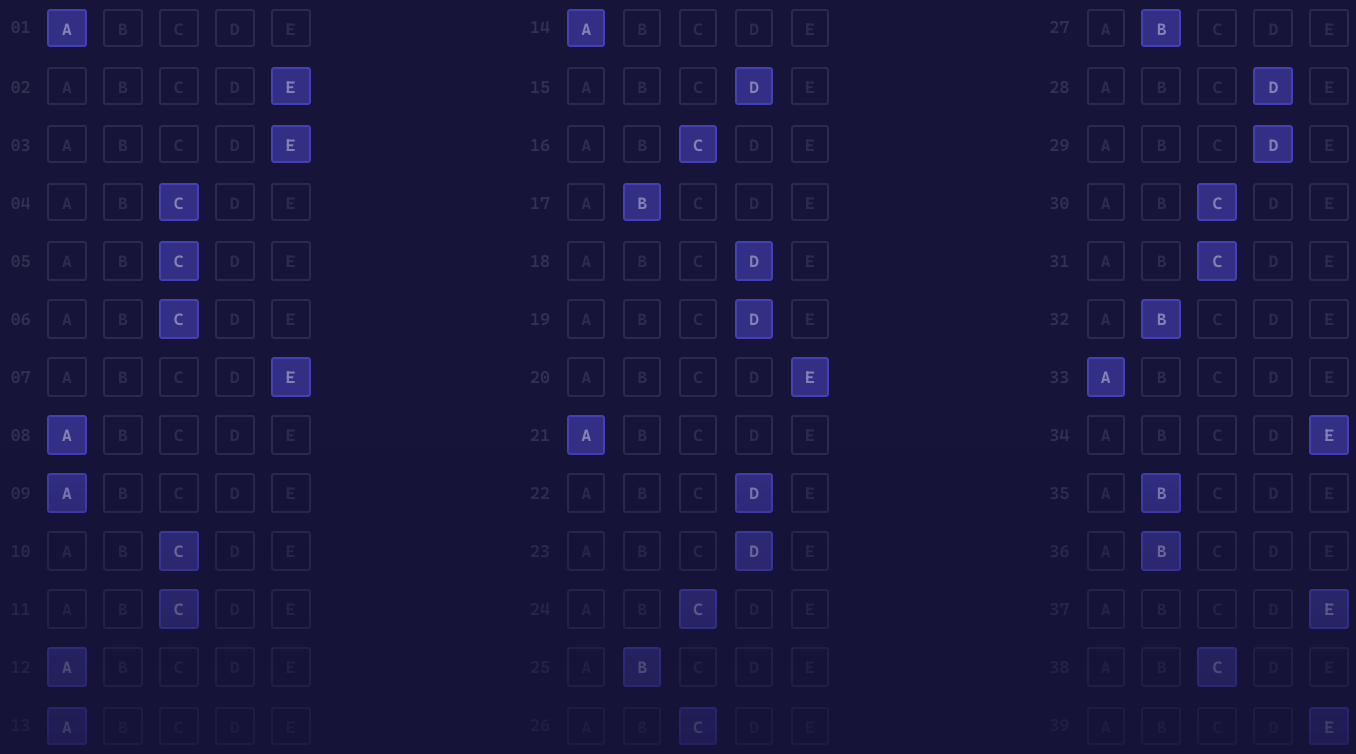
MOMETRIX
SMARTER PREP. BETTER RESULTS.



PRACTICE QUESTIONS

OutSystems

AS11P

Architecture Specialist O11 (PT)

EDITION	QUESTIONS	FORMAT
Demo	9	Limited Edition PDF

WEB www.mometrix.net

SUPPORT support@mometrix.net

Before you begin

What this is

9 practice questions for OutSystems `AS11P`, each with its answer key and an explanation. Answer a question before you read its key — the explanation is worth more once you have committed.

If something looks wrong

Send us three things and we will check it:

- the exam code, `AS11P`
- the question number
- the email address on your order

Write to **support@mometrix.net**. We reply within one business day.

mometrix.net

QUESTION 1 SINGLE CHOICE

An architect is reviewing a Traditional Web (ISP) application that has grown to over 200 eSpaces. Which architectural practice best supports long-term maintainability and reuse in OutSystems?

A Consolidate all logic into a single core eSpace that every other eSpace references directly.

B **Split the application into smaller, focused eSpaces and have them expose and consume functionality through published actions and referenced eSpaces.**

C Duplicate business logic across eSpaces so that each team can iterate independently.

D Move every reusable component to the database layer and avoid using eSpaces for shared logic.

ANSWER B

WHY OutSystems best practices promote small, focused, cohesive eSpaces that expose reusable functionality via published actions (Service Studio) and consume them through references. This keeps coupling explicit, ownership clear, and impact analysis manageable. Option A creates a monolithic eSpace that violates separation of concerns, Option C duplicates logic and breaks reuse, and Option D moves business logic away from the layer OutSystems is designed to manage.

mometrix.net

QUESTION 2 SINGLE CHOICE

A customer needs single sign-on between an OutSystems Reactive Web application and an existing corporate identity provider. Which OutSystems-supported integration pattern is recommended?

A Build a custom login screen that posts user credentials directly to the corporate database.

B Use the SAML 2.0 or OpenID Connect authentication providers available in OutSystems, configured against the corporate IdP.

C Embed the corporate IdP login page using an iframe and read the session cookie from JavaScript.

D Disable authentication in the OutUsers module to rely solely on the front-end web server.

ANSWER B

WHY OutSystems ships built-in authentication providers for SAML 2.0 and OpenID Connect in the Users module that can be configured to federate with external corporate identity providers, enabling SSO. Storing credentials directly in a database (A) and iframe-based scraping of sessions (C) are insecure and unsupported. Disabling OutSystems authentication (D) removes auditing and breaks the platform's user/session management.

mometrix.net

QUESTION 3 SINGLE CHOICE

An OutSystems application needs to publish domain events that several internal systems must react to without a tight point-to-point coupling. Which integration pattern should the architect prefer?

A A direct SOAP/REST call from the publishing eSpace to each consumer's endpoint.

B **An event-driven pattern using a lightweight message broker or OutSystems Timer queue that decouples producers from consumers.**

C A scheduled database polling job in each consumer that reads the publisher's tables.

D A shared file drop location that consumers pick up at fixed intervals.

ANSWER B

WHY Event-driven integration decouples producers from consumers so that adding or removing consumers does not require changes to the publisher. OutSystems supports asynchronous patterns through BPT processes, timers and external message brokers. Direct synchronous calls (A) couple availability of every consumer to the publisher, while tight DB polling (C) and shared files (D) are brittle, non-real-time patterns that are discouraged for system-to-system integration.

mometrix.net

QUESTION 4 SINGLE CHOICE

A customer plans to deploy an OutSystems Reactive Web application with high availability across two data centers. Which deployment topology is supported and recommended for active-active resilience?

A A single OutSystems server running both the front-end and the database in one zone, with a warm standby.

B Multiple OutRoles front-ends in a load-balanced farm across both data centers, backed by an active-active database cluster supported by the platform.

C Independent OutSystems installations in each data center that share no data and replicate only through nightly ETL jobs.

D A Docker container running the Controller and Service Studio on a developer laptop.

ANSWER B

WHY OutSystems supports high-availability topologies with multiple front-end servers (OutRoles) fronted by a load balancer and a clustered database tier, including active-active configurations across data centers when the database platform supports it. A single server (A) is not resilient. Disconnected installations with nightly ETL (C) do not deliver real-time HA. A developer laptop (D) is not a production deployment topology.

mometrix

QUESTION 5 SINGLE CHOICE

Two eSpaces in the same OutSystems application both need to read and write the same Customer entity. What is the recommended approach to keep the data layer consistent?

A Duplicate the Customer entity in each eSpace and use database triggers to synchronize them.

B **Centralize the Customer entity in a single eSpace and expose CRUD operations as published actions consumed by the other eSpaces.**

C Allow each eSpace to read the Customer table directly via advanced SQL, bypassing the entity.

D Store the Customer data in a flat file inside the file system and let each eSpace parse it.

ANSWER B

WHY OutSystems recommends centralizing entities in their owning eSpace and exposing data operations through published actions or referenced entities, which keeps a single source of truth, enforces security rules and lets the platform manage the schema. Duplicating entities (A) breaks referential integrity. Direct advanced SQL bypassing the entity (C) bypasses OutSystems' permission engine and risks schema-drift. Flat files (D) are not a valid persistent data layer for transactional entities.

mometrix.net

QUESTION 6 SINGLE CHOICE

An architect needs to ensure that all unhandled exceptions across an OutSystems Reactive Web application are captured centrally with correlation IDs for support. Which approach is recommended?

A Wrap every screen action with a local try/catch and write to a free-text log file on the front-end server.

B **Implement a global exception handler that logs to the OutSystems Logging database (Log_General_Error) and to an external monitoring tool via a REST endpoint.**

C Disable error logging and rely on end users to email screenshots when something fails.

D Use only client-side console.log() output in the browser.

ANSWER **B**

WHY OutSystems provides a global exception handler mechanism that can capture exceptions with their request context and write them to the built-in Log_General_Error tables, and architects commonly forward them to an enterprise monitoring/observability tool. Per-screen try/catch (A) is brittle and easily missed; disabling logging (C) and client-side console output (D) do not give operations-grade visibility.

mometrix

QUESTION 7 SINGLE CHOICE

Which combination correctly describes the security practices an OutSystems Architecture Specialist should apply by default in a production deployment?

A Allow anonymous access to every screen and disable the platform's built-in role-based permissions for performance.

B Use HTTPS only between the load balancer and the browser, but keep internal traffic on HTTP between front-ends and the database.

C Enforce HTTPS end-to-end, keep the platform and database servers on a private network, encrypt sensitive data at rest, and use role-based access with least privilege.

D Store database credentials in plain text on the front-end servers so they can be quickly rotated by operations.

ANSWER C

WHY Production OutSystems deployments must enforce HTTPS end-to-end, keep infrastructure on private networks, protect sensitive data at rest (e.g. database encryption), and apply role-based, least-privilege access through the Users module. Disabling anonymous checks and roles (A), downgrading internal transport to HTTP (B), and storing credentials in plaintext (D) are all insecure and contrary to platform guidance.

mometrix.net

QUESTION 8 SINGLE CHOICE

A Reactive Web application is experiencing slow page loads because of expensive read-only reference data lookups performed on every screen render. What is the recommended architectural optimization?

A Increase the front-end heap size of every front-end server indefinitely.

B Cache the reference data in memory at application startup using a singleton with a sensible refresh strategy and remove it from per-request flows.

C Remove caching entirely and rely on the database to serve every lookup.

D Hard-code the reference data as static values inside every screen.

ANSWER B

WHY Caching expensive, stable reference data at the application layer (e.g. a singleton loaded at startup with a refresh strategy) is a standard OutSystems performance pattern that reduces database round trips and improves screen render times. Tuning heap size (A) treats a symptom rather than the cause, removing caching (C) ignores a valid optimization, and hard-coding values (D) prevents updates and breaks maintainability.

QUESTION 9 MULTIPLE CHOICE

Which components are typically part of an OutSystems application's logical architecture? (Choose two.)

A eSpaces (now Modules) that encapsulate the UI, logic, and data of a bounded context

B Processes that orchestrate long-running, stateful business flows through human or system activities

C A built-in proprietary message-oriented middleware replacing the need for any external broker

D Timers that are triggered exclusively by external cron expressions evaluated outside the platform

ANSWER A · B

WHY OutSystems applications are built around Modules (formerly eSpaces) that group UI, logic, and data for a bounded context, and Processes (BPT) that coordinate long-running, multi-step business activities involving humans and systems. There is no proprietary message-oriented middleware replacing external brokers — integration uses standard connectors and APIs (option C is incorrect). Timers run on the OutSystems scheduler and can be configured with simple/advanced recurrence, they do not require an external cron expression evaluated outside the platform (option D is incorrect).