



**Free Questions for [Architecture-Specialist-11](#) by [certsdeals](#)**

**Shared by [Salas](#) on [15-04-2024](#)**

**For More Free Questions and Preparation Resources**

**[Check the Links on Last Page](#)**

# Question 1

---

## Question Type: MultipleChoice

---

Which of the below matches the most to Core Module Pattern - Base ECS Pattern...

### Options:

---

- A-** ... a wrapper used to contain the logic, actions and data that will expose code that is inside of external library or to inspect external database and import the data structures so they can be used as entities inside of OS
- B-** ... caches only summary data that is frequently listed, joined or searched. Full detail for a single entry is fetched directly from external system. Use when whole database too big or costly to synchronize. Details are only required for single entities (not lists)
- C-** Same as ECS with local replica but synchronization logic is separated. Pro: Code independence. Consumers of CS is not affected by Sync. Sync can orchestrate several CS
- D-** ... is a pattern with two modules, a connector module that can be used to encapsulate an external API with the input/output structures and a wrapper module to expose the normalized API to the consumers.
- E-** ... is needed if data is coming from MULTIPLE external systems. IS will decide which driver to use depending on the data.
- F-** Same as ECS with local replica but API module is provided. So any changes to the external system can notify OS, which OS then gets update from the ERP system (subscription system)
- G-** Same as Base ECS pattern, but have a local replica. Store data to serve as a local cache. Pro: Leverage Entity Use, Simpler Integration API. Con: Less impact on source system

**H-** ... Entity is not in Outsystems but in an external ERP system. IS just makes remote call to external system/database. No data is being kept inside OS. Data retrieval may not be optimized as it needs to traverse two different systems to get the information back. Con: Integration API must support all use cases

**I-** ... tries to fetch data from local cache entity, if not there, get single entry from the external system. Cache only that record (read-through caching) Use when whole database too big or costly to synchronize. Integration only touches a small portion of the database. Avoid if access to lists of data is needed up front

**J-** ... Entity is exposed as read-only and API is available to centralize business logic for entity creation/update

**Answer:**

---

D

## Question 2

---

**Question Type: MultipleChoice**

---

OAPI is different compared to API. What is inside an OAPI?

**Options:**

---

**A-** Service Action

**B-** REST API

**Answer:**

---

A

## Question 3

---

**Question Type:** MultipleChoice

---

ISO/IEC 25010:2011 adds two more aspect to ISO/IEC 9126:1991. Which of the below is not part of it?

**Options:**

---

**A-** Compatibility : Compatibility was also added as a way to ensure that a piece of software can work together with other systems

**B-** Security : Security ensure software solutions can protect information and data

**C-** Integration : Integration to allow data exchange with other software

**Answer:**

---

C

## Question 4

---

**Question Type:** MultipleChoice

---

Which of the following recommendations should be applied to the Core layer?

### Options:

---

- A- Core modules should have Front-end Screens for testing purposes.
- B- Core modules should not have Core Entities.
- C- Core modules should have public Read-only Entities.
- D- Core modules should not have business logic.

### Answer:

---

C

## Question 5

---

**Question Type:** MultipleChoice

---

Which of the below is NOT a suitable advice for designing a LIGHTWEIGHT mobile local storage?

**Options:**

---

- A- Denormalize the local storage
- B- Define only used attributes
- C- Group relevant information
- D- Store all records no matter if relevant or irrelevant

**Answer:**

---

D

## Question 6

---

**Question Type: MultipleChoice**

---

Which of the below is not part of the Architecture Validation Rules?

**Options:**

---

- A- Layer Applications Correctly
- B- Don't Mix Sponsors : Split the Apps based on the Lines of Business
- C- Layer Entities Correctly
- D- Layer Modules Correctly
- E- Don't Mix Owners

**Answer:**

---

C

## Question 7

---

**Question Type: MultipleChoice**

---

"Splitting UI elements per functionality" is a best practice to avoid monolithic mobile UI modules? Which of the below is not a best practice?

**Options:**

---

- A- Screens as layouts that have placeholders and load content from blocks. The blocks come from CW modules.
- B- Include the Common flow and Menu artifacts
- C- Put CSS in blocks or screens. Allows better performance and seamless screen transitions
- D- Blocks grouped in functional modules with independent life cycles
- E- CSS should only be placed in the mobile theme module

**Answer:**

---

C

## Question 8

---

**Question Type:** MultipleChoice

---

Considering Discovery, which of the following sentences is FALSE?

**Options:**

---

- A- Discovery automatically assigns a module to an Architecture Canvas layer, following the module's naming convention.
- B- Discovery only validates architecture rules at the module level.



- C-** Discovery allows you to determine for every module, which elements are being consumed by other modules.
- D-** Discovery cannot categorise modules to domains

**Answer:**

---

B

## Question 9

---

**Question Type:** MultipleChoice

---

Which of the below are FALSE about UI framework?

**Options:**

---

- A-** Patterns - are blocks that provide common and reusable UI. Chat message, date picker and carousel are a few examples. Contains reusable UI components.
- B-** All of the above are true
- C-** Theme - for the look and feel of your applications, which includes screen layouts, global stylesheet and grid definitions. Contains CSS styles and classes.
- D-** Templates - Defines the structure of the application modules. Templates contains screen p definitions and basic set of functionalities,

including authentication.

Contains common layouts, menu, login and reference to the theme .

**Answer:**

---

B

## Question 10

---

**Question Type:** MultipleChoice

---

Architecture Canvas has 3 layers. Which is not part of the 3 layer framework?

**Options:**

---

A- Foundation

B- Ochestrator

C- End-User Layer

D- Core

**Answer:**

---

B

**To Get Premium Files for Architecture-Specialist-11 Visit**

**<https://www.p2pexams.com/products/architecture-specialist-11>**

**For More Free Questions Visit**

**<https://www.p2pexams.com/outsystems/pdf/architecture-specialist-11>**

