



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

Shared by [Wagner](#) on [29-01-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

_CW module is for

Options:

- A-** Logic to Synchronize data in CS's with an external system. Isolating this logic makes the CS completely system agnostic and it's easier to decouple or replace the external system.
- B-** A BL becomes a Calculation Engine if it performs complex calculations, (e.g. an invoice calculation engine or an insurance simulator). Engines are usually subject to versions.
- C-** Technical wrapper to expose an API to External consumers, keeping core services system agnostic and supporting multiple versions of the API.
- D-** Core Widgets (blocks), to manage complexity, composition or to have its own lifecycle.
- E-** Reusable Core Services with public entities, actions, and blocks.

Answer:

D

Question 2

Question Type: MultipleChoice

What is the new name for Architecture Dashboard?

Options:

- A- Architecture Validation
- B- Architecture Canvas
- C- Architecture Framework
- D- AI Mentor

Answer:

D

Question 3

Question Type: MultipleChoice

Which of the below matches the most to Core Module Pattern - ECS with Local Replica Pattern...

Options:

- A-** ... Entity is not in Outsystems but in an external ERP system. IS just makes remote call to p 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
- B-** ... is a pattern with two modules, a connector module that can be used to encapsulate an O external API with the input/output structures and a wrapper module to expose the normalized API to the consumers.
p 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
- C-** ... 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
- D-** ... 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)
- E-** ... Entity is exposed as read-only and API is available to centralize business logic for entity creation/update
- F-** 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
- G-** ... 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

H- ... is needed if data is coming from MULTIPLE external systems. IS will decide which driver to use depending on the data.

I- 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)

Answer:

G

Question 4

Question Type: MultipleChoice

Which of the following is a benefit of having good architecture?

Options:

A- Poor service abstraction

B- Unmanageable dependencies

C- Manages complexity

D- Slow-moving legacy systems

Answer:

C

Question 5

Question Type: MultipleChoice

What is the common naming convention for a mobile version of a CS module?

Options:

A- M_CS

B- _CS

C- mobile_CS

D- m_cs

E- m_CS

Answer:

E

Question 6

Question Type: MultipleChoice

What does NOT happen due to a lack of architecture concerns?

Options:

- A-** Inflexible and slow-moving legacy systems : adapting legacy systems to business changes may be difficult. Changes in complex and inflexible systems can take a long time
- B-** Tech Debt : AI Mentor will raise architectural tech debt such as cyclic dependency and side to side dependency
- C-** Unmanageable dependencies : System not isolated from each other. Updating or replacing a system has a cascade/snowball effect on other systems
- D-** Poor service abstraction : Business concepts not correctly isolated, business rules tend to be spread over different systems and little to no code reuse

Answer:

B

Question 7

Question Type: MultipleChoice

What is NOT a best practice for Mobile Application Architecture: Local Storage?

Options:

A- Adopt the correct sync frequency: Either at process start and online or at process/transaction end and online

B- Sync on every screen or online event

C- Sync data required per use case: Sync summary data on Session start. On data selection, sync p its details.

Example: if you are accessing Purchasing data, sync the summary data on Session start. On selecting the Purchasing item, sync the details of the Purchasing item such as photos or price, a

Answer:

B

Question 8

Question Type: MultipleChoice

What is NOT a best practice for Mobile Application Architecture: transactions & granularity?

Options:

A- Have long synchronizations in a single transaction. Better UX as app does not need to sync all the time. Is prepared for constant offline or device standby

B- Ensure order and sync granularity. Sync incrementally by entity with partial commit. This way O synchronizations is prepared for constant interruptions and allow retries without repeating the entire synchronization from the start.

Answer:

A

Question 9

Question Type: MultipleChoice

What is Application Composition Process?

Options:

A- Answer is the above.

B- Process to package modules into apps

Answer:

B

Question 10

Question Type: MultipleChoice

Which of the below matches the most to Core Module Pattern - ECS Summary Cache only variation

Options:

A- 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

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- 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)

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

F- ... is needed if data is coming from MULTIPLE external systems. IS will decide which driver to use depending on the data.

G- ... tries to fetch data from local cache entity, if not there, get single entry from the external p 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.

H- ... 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

I- ... 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

J- ... 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.

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>

