



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

Shared by [Moody](#) on [12-12-2023](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

Which is not a reason you should use Architecture Canvas?

Options:

- A- Promotes segregation and loose coupling of services
- B- Optimizes lifecycle independence
- C- Promotes abstraction of reusable services
- D- Minimizes impact of changes

Answer:

A

Question 2

Question Type: MultipleChoice

Which of the below is NOT part of the three step process of Architecture Design Process?

Options:

A- Assemble

B- Shape

C- Disclose

D- Organize

Answer:

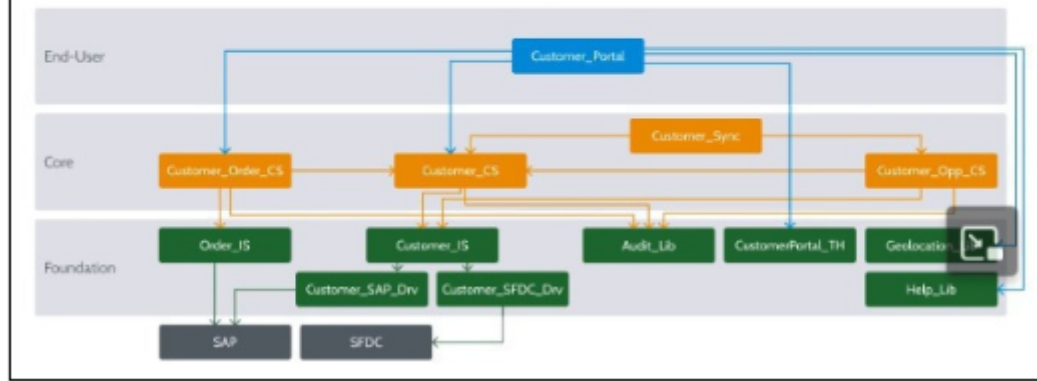
B

Question 3

Question Type: MultipleChoice

What is the Architecture Blueprint?

Architecture Reference Map



Options:

A- Is the end result of Architecture Design Process. At this point you should add dependencies between the Modules.

B- Above is the answer

Answer:

A

Question 4

Question Type: MultipleChoice

_Sync module is for

Options:

- A-** Reusable Core Services with public entities, actions, and blocks.
- B-** Isolated Business Logic (Actions) or Core Widgets (blocks), to manage complexity, composition or to have its own lifecycle.
- C-** 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.
- D-** 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.
- E-** Technical wrapper to expose an API to External consumers, keeping core services system agnostic and supporting multiple versions of the API.

Answer:

D

Question 5

Question Type: MultipleChoice

_Eng module is for

Options:

- A-** Isolated Business Logic (Actions) or Core Widgets (blocks), to manage complexity, compositio or to have its own lifecycle.
- B-** 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.
- C-** 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.
- D-** Technical wrapper to expose an API to External consumers, keeping core services system agnostic and supporting multiple versions of the API.
- E-** Reusable Core Services with public entities, actions, and blocks.

Answer:

C

Question 6

Question Type: MultipleChoice

There are 3 common scenarios for Sharing a Style Guide. Which of the below is not part of the scenario.

Options:

- A-** Intranet (Single Sign On): Own Menu, Common Login Flow. Menu is defined in the Custom Template, but Login is defined in the Custom Theme. Application reference the Custom Them which picks up the Login.
- B-** Enterprise Apps: Common Menu, Common Login Flow. Menu is defined in the Custom Theme, but Login is defined in the Custom Template.
- C-** Independent Apps : Own Menu, Own Login Flow. Login and Menu is defined in the Custom Template. Applications reference to its own Application Theme thus do not use the Login and Menu in the Custom Template.
- D-** Portal : Common Menu, Common Login Flow. Login and Menu is defined in the Custom Them Application Theme reference the Custom Theme thus have a shared menu and login flow.

Answer:

B

Question 7

Question Type: MultipleChoice

Which of the following options denotes the advantages of defining a Style Guide up front?

Options:

- A- Security and scalability.
- B- Speed up the development phase.
- C- Improve performance and maintainability.
- D- Allows apps and the Style Guide to be deployed to Production.

Answer:

C

Question 8

Question Type: MultipleChoice

Which of the below matches the most to Core Module Pattern - ECS with Direct Integration 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-** ... 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-** ... 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-** 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
- E-** ... 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.
- F-** ... Entity is exposed as read-only and API is available to centralize business logic for entity creation/update
- G-** . 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
- H-** 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)

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

J- ... 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

Answer:

H

Question 9

Question Type: MultipleChoice

_Lib module is for

Options:

A- have several integration services with different systems, performing the same type of O operation (e.g. printers) you can create several drivers exposing the same API, with specialized implementations (like the transparency services pattern).

B- Technical wrapper to consume and normalize an external service

- C-** Reusable Core Services with public entities, actions, and blocks.
- D-** Theme, look & feel elements, menu, etc.
- E-** Generic Library module
- F-** Reusable UI Patterns for layout and display only - no Business logic.

Answer:

E

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>

