

Practice Exam for Senior System Architects

The Certified Pega Business Architect version '24.2 exam consists of 60 questions. Below is a sample.

Question 1

A development team is building a Pega application for a financial services company. The application handles highly sensitive customer data, including social security numbers (SSN) and credit card details, which must be protected both when stored in the database and when visible on the Clipboard. The team also needs to ensure that this data is not exposed in logs or search indexes.

Which of the following encryption approaches should the System Architect configure to meet these requirements? (Choose all that apply.)

- A. Implement a custom cipher and key combination for a hybrid encryption model.
- B. Use an irreversible hashing process for the sensitive properties.
- C. Apply class-level (BLOB) encryption to the case type to protect the entire record at rest.
- D. Configure a PropertyEncrypt Access Control Policy for the specific sensitive properties.
- E. Enable automatic key rotation as an advanced configuration.

Question 10

A System Architect is tasked with securing an existing Pega application. The application's database schema has been optimized for reporting by exposing several columns that contain sensitive customer information. However, this exposed data is not encrypted, creating a security vulnerability. The team needs to encrypt this sensitive data while still allowing it to be used in reports and other Pega functions.

Which of the following actions should the System Architect take to remediate this vulnerability? (Choose all that apply.)

- A. Configure automatic key rotation to ensure the exposed data is encrypted.
- B. Implement class-level (BLOB) encryption, which automatically encrypts all exposed properties.
- C. Use property-level encryption, as it can encrypt properties outside of the BLOB, including exposed columns.
- D. Configure a PropertyEncrypt Access Control Policy on the specific properties to encrypt them in the database, Clipboard, and reports.
- E. Change the property type to TextEncrypt, which is the standard practice for encrypting sensitive data.

Question 20

An organization needs to create a series of reports to monitor the efficiency of its Loan Application process. The business is interested in two key areas:

1. Identifying the average time it takes for a loan application to be approved.
2. Tracking the number of Loan Applications that are submitted for a specific loan type.

Which two types of metrics and which two classes would a Pega System Architect use to fulfill these requirements? (Choose all that apply.)

- A. Business Metrics for the number of loan applications.
- B. Process Metrics for the average time to approve a loan application.
- C. The History class to track the number of loan applications.
- D. The Work class for the number of applications and the History class for the time to approve.
- E. The Assign-Worklist class for the number of applications and the History class for the time to approve.

Question 30

A Pega System Architect is tasked with integrating a new Claims application with a custom, non-native front-end. The business requires the integration to be highly performant, with minimal network traffic and a rich data payload for a single request. The team is considering using the Pega DX API.

Which two of the following are benefits of using the **Constellation DX API** over the Traditional DX API in this scenario? (Choose two.)

- A. The Constellation DX API allows for the use of traditional Pega harnesses and sections to build the user interface.
- B. The Constellation DX API has a seven-times smaller network payload and 30% fewer requests on first load.
- C. The Constellation DX API responses are less robust and require additional API calls for validation and actions.
- D. The Constellation DX API provides a set of model-driven API endpoints that can be used by both native and non-native UIs.
- E. The Constellation DX API relies on an entirely different JSON structure that separates data from the View layout.

Question 40

A Pega System Architect is troubleshooting a failing connector that integrates with a third-party Shipping service. The error occurs when the service returns an invalid response format that the Pega connector cannot parse. The architect needs to implement a robust error handling mechanism to address this issue.

Which two of the following are best practices for handling this type of integration error? (Choose two.)

- A. Use the `pxErrorHandlingTemplate` data transform to create a reusable data transform for detecting errors in the response.
- B. The `ConnectionProblem` flow should be configured to route the work item to a workbasket and notify an operator about the issue.
- C. Write the error details, including the connector request and response, to the log file to aid in troubleshooting.
- D. Rely solely on the Error Handling Flow to detect and address all types of integration errors, as it is always enabled.
- E. Configure a when condition in the Response Data Transform to check for specific error messages and then apply the reusable error handling data transform.

Question 50

A Pega developer is using an activity to automate a complex business process. The activity needs to call another activity and then return control to the original calling rule after the second activity is completed. Which step instruction should the developer use in the first activity to achieve this behavior?

- A. Branch
- B. Call
- C. Jump
- D. Fork
- E. Queue

Question 60

A Pega application requires a batch process that reads a large number of work objects from the database, updates their status, and performs some business logic. This process is time-consuming and needs to be robust enough to handle system restarts without corrupting data. What are the best practices for implementing this background process? (Choose all that apply.)

- A. Break down large work items into smaller, individually processed work items.
- B. Use a Queue Processor as it is the most suitable for tasks that recur on a regular basis.
- C. Implement checkpoints by saving the state of jobs in persistent storage or a message queue.
- D. Use a Job Scheduler and a single activity to perform all the logic.
- E. Use the Obj-Save and Commit methods inside a loop to save each record as it is processed.

Answer Key

Question 1

- **Correct Answers: D and C**
- **A. Incorrect.** While Pega uses a cipher and key, the question asks for the correct Pega Platform approach, not a custom solution. Pega's built-in encryption mechanisms are the correct path.
- **B. Incorrect.** Hashing is an irreversible process primarily used for passwords, not for data that needs to be decrypted and viewed by authorized users.
- **C. Correct.** Class-level (BLOB) encryption is an efficient method to encrypt an entire Case or Data Record stored as a Binary Large Object in the database. It protects the data at rest, which is a key requirement.
- **D. Correct.** Property-level encryption is implemented by configuring a PropertyEncrypt Access Control Policy. This approach is specifically designed to encrypt properties in and outside the database, including the Clipboard, logs, search indexes, and reports. This directly addresses the requirement to protect data in these specific locations.
- **E. Incorrect.** Automatic key rotation is a best practice for security but is an advanced configuration and does not, by itself, determine the encryption approach.

Question 10

- **Correct Answers: C and D**
- **A. Incorrect.** Automatic key rotation is a security best practice, but it does not determine the encryption method itself or whether exposed columns are encrypted.
- **B. Incorrect.** Class-level (BLOB) encryption specifically does **not** encrypt properties that are exposed as columns for reporting. This approach would not solve the problem.
- **C. Correct.** Property-level encryption is the correct approach because it encrypts properties both inside and outside the database, including properties that are optimized for reporting.
- **D. Correct.** Property-level encryption is implemented by configuring a PropertyEncrypt Access Control Policy. This policy encrypts the data in various locations, including the database, Clipboard, logs, and search indexes.
- **E. Incorrect.** The TextEncrypt property type is deprecated, and the correct approach is to configure a PropertyEncrypt Access Control Policy.

Question 20

- **Correct Answers: B and D**
- **A. Incorrect.** Business Metrics would be used for the number of loan applications, which is a measure of the success or failure of a business process, but this answer only provides half of the correct information. The question asks for both types of metrics and the associated classes.
- **B. Correct.**
Process Metrics measure "how work is performed". The time needed to complete an assignment is a process metric. The average time to approve a loan application is a process metric.
- **C. Incorrect.** The History class is used to measure performance and idle time. The Work class stores the case itself, and it is the correct place to find the number of applications.
- **D. Correct.** The Work class contains the case data, so a report on this class can count the number of Loan Application cases created. The History class contains performance data, such as time spent on tasks, which would be used to calculate the average time to approve.
- **E. Incorrect.** Assign-Worklist is used for assignments for a specific user. The number of applications would be found in the Work class.

Question 30

- **Correct Answers: B, D, and E**
- **A. Incorrect.** The Constellation architecture, which the DX API is part of, is a shift away from traditional harnesses and sections towards a model-driven approach.
- **B. Correct.** The Constellation DX API "has drastically reduced networked traffic with 30% fewer requests on first load and a seven-times smaller network payload," which directly addresses the requirement for a highly performant integration with minimal network traffic.
- **C. Incorrect.** The document states that the "DX API responses are much more robust, containing all the necessary information to reproduce Views in any front-end framework, including layouts, fields, validation, conditional visibility, and actions."
- **D. Correct.** The Constellation DX API "supports both Pega native UIs and non-native UIs," which is a key requirement for the scenario of integrating with a custom front-end. The "same API endpoints power Pega software's out-of-the-box user experiences and its custom front-end experiences."
- **E. Correct.** The document explicitly states that the Constellation DX API "is an entirely re-engineered experience" and has "an entirely different JSON structure, which clearly separates data from the View layout".

Question 40

- **Correct Answers: C and E**
- **A. Incorrect.** The `pxErrorHandlingTemplate` is a reusable data transform, but it is typically invoked by another rule (like a Response Data Transform), not used to directly detect the error. The error detection logic is configured in the calling rule.
- **B. Incorrect.** The `ConnectionProblem` flow is the default error handling flow, but a flow is generally invoked for errors not detected by a data transform. In this scenario, the issue is an invalid response format, which is typically handled in a data transform. A more specific, custom flow might be needed, but the default flow would be for connection issues.
- **C. Correct.** Writing detailed error information, including the request and response, to the log file is a critical best practice for troubleshooting integration issues. This helps developers identify patterns and the root cause of the error.
- **D. Incorrect.** While the Error Handling Flow is always enabled, it is not the best practice to rely on it for all errors. For errors that can be detected early (e.g., in a Response Data Transform), handling them there provides a better user experience and avoids the overhead of a flow.
- **E. Correct.** Configuring a when condition in the Response Data Transform to check for specific error messages or a malformed response and then calling a reusable error handling data transform is a standard best practice for handling integration errors.

Question 50

- **Correct Answer: B**
- **A. Branch** This is incorrect. Branch runs the specified activity but returns control to the rule that *originally called* the first activity. The first activity terminates.
- **B. Call** This is the correct answer. The Call step instruction executes the specified activity and then returns control back to the calling activity upon completion, allowing the first activity to continue its own processing.
- **C. Jump** This is incorrect. Jump is a step instruction that transfers control to a different step within the *same* activity, not to a different activity.
- **D. Fork** This is incorrect. Fork is not a standard Pega activity method.
- **E. Queue** This is incorrect. Queue is a method that is part of the Queue-For-Processing instruction, used for asynchronous processing. It does not return control to the calling activity synchronously.



Question 60

- **Correct Answers:** A, C
- **A. Break down large work items into smaller...** This is a correct answer. For a robust background process, Pega best practice is to break down large tasks into smaller, more manageable ones. This prevents a single failure from affecting the entire process and makes it easier to recover.
- **B. Use a Queue Processor...** This is incorrect. A Queue Processor is for processing individual tasks that are queued. For a regularly scheduled batch job, a Job Scheduler is the appropriate tool.
- **C. Implement checkpoints by saving the state of jobs...** This is a correct answer. Checkpointing is a crucial best practice for long-running background processes. By saving the state of the job, the process can gracefully handle restarts or failures without corrupting data or starting over from the beginning.
- **D. Use a Job Scheduler and a single activity...** This is incorrect. While a Job Scheduler is the right tool, using a single activity for all logic is a poor practice. Best practice dictates separating business logic from transaction logic to ensure better maintainability and error handling.
- **E. Use the Obj-Save and Commit methods inside a loop...** This is incorrect. Using Obj-Save and Commit repeatedly in a loop is a performance anti-pattern. It can lead to database locking issues and is inefficient. The background processing features are designed to handle transactions more effectively.