Assessment

Part 1: Leadership Scenarios:

Scenario 1: Navigating Technical Disagreement and Conflict

1. How would you intervene?

I would intervene immediately and respectfully de-escalate the conversation by pausing the meeting and acknowledging both Alex and Ben's concerns. I'd emphasize that both viewpoints have merit and that our goal is to collaboratively choose the best approach for our current context.

2. Steps to facilitate a constructive discussion:

- Schedule a dedicated follow-up session for deeper technical evaluation.
- Create an evaluation matrix with key criteria (scalability, team capability, delivery speed, long-term maintenance).
- Encourage data-backed proposals: pros/cons, PoCs, or benchmarks.
- Facilitate input from all team members to balance the discussion and encourage psychological safety
-

3. Ensuring team cohesion:

- Reiterate that disagreement is natural and even healthy in engineering teams.
- Communicate that the final decision will be made based on project needs, not egos.
- Highlight mutual respect and reinforce team values.
- After decision, organize a short retrospective to reflect on how the conflict was handled.

Scenario 2: Addressing Underperformance and Motivation

1. Approach to Chris:

- I'd invite Chris to a one-on-one conversation in a private, empathetic setting. I'd avoid judgment and begin by expressing concern and support, not criticism.

2. Understanding root cause:

- Use open-ended questions to allow Chris to share.
- Offer flexibility or resources (e.g., time off, professional help) if personal issues are involved.
- Observe if it’s workload burnout, lack of challenge, or unclear expectations.

3. Balancing support and productivity:

- Reallocate or reduce Chris’s workload temporarily.
- Communicate transparently with the team (without sharing personal details).
- Involve HR if extended support is required.

4. Monitoring and support:

- Set short-term, achievable goals with Chris.
- Offer regular check-ins (weekly/bi-weekly).
- Provide mentorship or pair programming to re-engage.
- Document progress and support provided.

Scenario 3: Managing Scope Creep and Shifting Priorities

1. Immediate action:

- Request a quick sync with the Product Owner to understand the regulatory requirement, timeline, and priority.
- Freeze any new development temporarily.

2. Communicating to team:

- Be transparent and explain why the change is necessary (compliance).
- Acknowledge current efforts and pressure.
- Frame this as a collaborative challenge and invite input on reprioritization.

3. Re-prioritizing and stakeholder management:

- Conduct impact analysis of changing scope.
- Involve Scrum Master and Product Owner to re-groom backlog.
- Communicate new timelines to stakeholders with justification.

4. Empowering the team:

- Involve the team in decision-making.
- Provide clear goals and boundaries.
- Celebrate small wins and ensure support through the transition.

Scenario 4: Introducing a New Technology/Process

1. Advocating for the new tool:

- Share a demo or case study of how the new framework improves speed or quality.
- Tie the adoption to team pain points (e.g., current tool limitations).

2. Overcoming resistance:

- Acknowledge concerns and ask for feedback.
- Identify champions in the team to try it first.
- Offer opt-in pilots instead of forced adoption initially.

3. Supporting transition:

- Schedule onboarding/training sessions.
- Allocate spike time in the sprint for learning.
- Provide ongoing documentation and pairing.

4. Measuring success:

- Track metrics like test coverage, defect leakage, time to execute tests.
- Gather team feedback after a few sprints.
- Compare regression cycle times before and after adoption.

Scenario 5: Leading a Post-Mortem of a Major Incident

1. Preparing the meeting:

- Gather facts: timelines, logs, incident details.
- Set the tone by sending out a blameless post-mortem agenda.

2. Meeting objectives:

- Identify what went wrong, how it was detected, how we responded, and how to prevent recurrence.
- Discuss both technical and process improvements.

3. Guiding root cause analysis:

- Use "Five Whys" or fishbone diagrams.
- Focus on systems thinking, not individual failure.
- Encourage psychological safety for open discussion.

4. Follow-up actions:

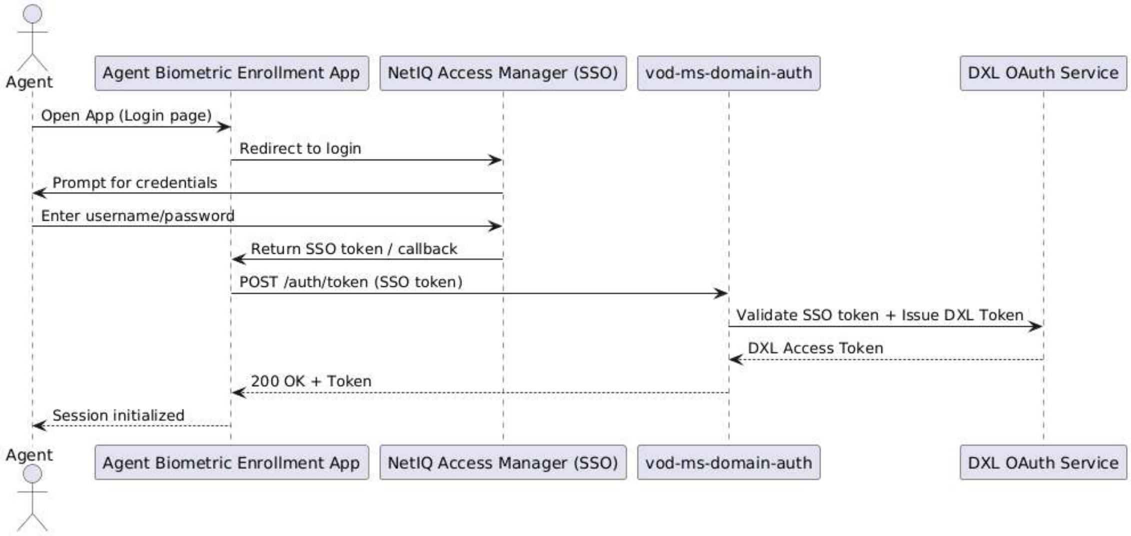
- Document action items with owners and deadlines.
- Schedule a review to ensure items are completed.
- Share lessons learned with the broader team/org.

Part 2: Architectural Assessment:

1. Login Endpoint – PlantUML Sequence Diagram

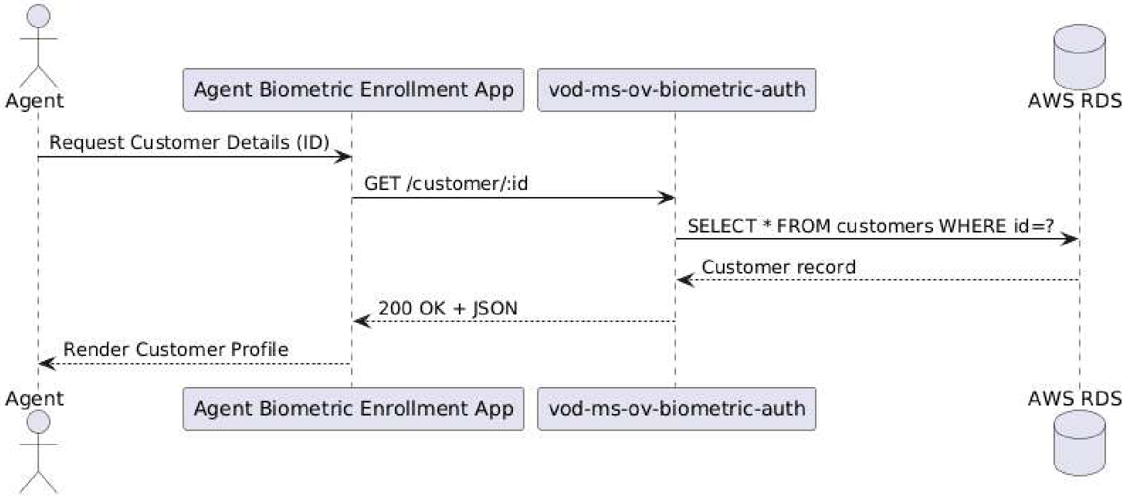
This involves:

- Authentication via NetIQ (SSO)
- Token issuance via vod-ms-domain-auth (DXL OAuth)
- Access token returned to the frontend



2. Customer Details Endpoint – PlantUML Sequence Diagram

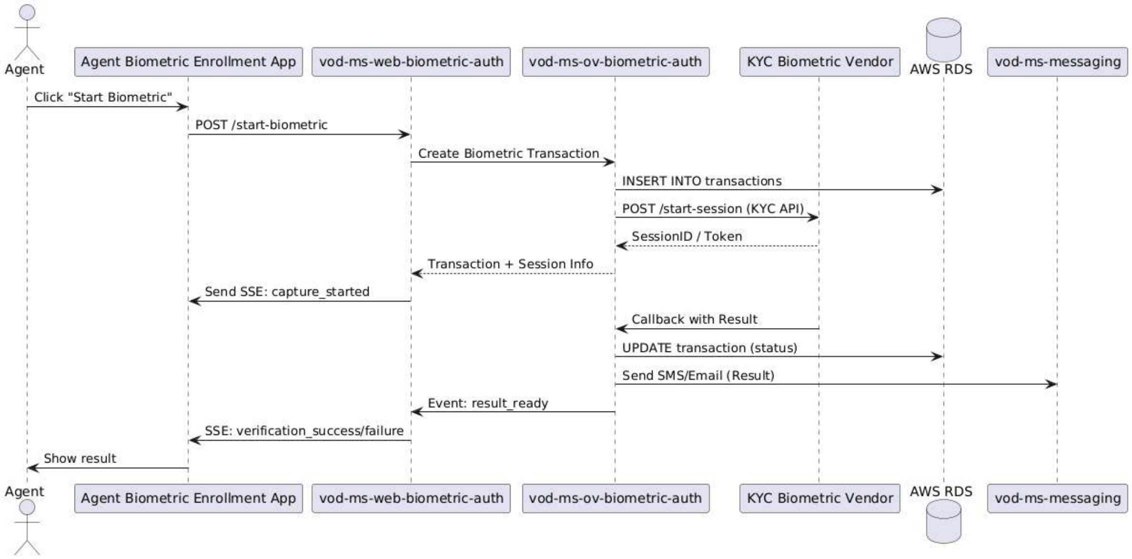
- This represents retrieving customer or biometric session details.



3. Biometric Transaction Init – PlantUML Sequence Diagram

This involves:

- Frontend initiating a biometric capture
- Backend triggering KYC vendor flow
- Receiving callback and finalizing verification
- SSE updates sent to frontend


Agent
Agent Biometric Enrollment App
vod-ms-web-biometric-auth
vod-ms-ov-biometric-auth
KYC Biometric Vendor
AWS RDS
vod-ms-messaging
Click "Start Biometric"
POST /start-biometric
Create Biometric Transaction
INSERT INTO transactions
POST /start-session (KYC API)
SessionID / Token
Transaction + Session Info
Send SSE: capture_started
Callback with Result
UPDATE transaction (status)
Send SMS/Email (Result)
Event: result_ready
SSE: verification_success/failure
Show result
Agent
Agent Biometric Enrollment App
vod-ms-web-biometric-auth
vod-ms-ov-biometric-auth
KYC Biometric Vendor
AWS RDS
vod-ms-messaging