Here is the rewritten **Orgo v2 Blueprint Section 6 - Workflow Integration** with the requested corrections applied:

### Section 6: Workflow Integration

This section outlines how workflows are designed, implemented, and managed in Orgo. Updates include modularized workflow logic using YAML templates for consistency and ease of customization.

### 6.1 Purpose of Workflow Integration

**Objective**:

* Define and implement structured workflows for common and sensitive use cases.
* Ensure smooth automation, traceability, and compliance.

**Outcome**:

* Tailored workflows for specific organizational needs, enabling efficient email routing, task management, and escalation.

### 6.2 Workflow Design Principles

#### Modular and Customizable Workflows

Workflows are defined in YAML or JSON, allowing for easy adaptation to organizational requirements.

**Example YAML Rule**:

workflow\_rules:

- type: "maintenance"

condition: "subject contains 'leak'"

action:

route\_to: "maintenance@organization.com"

attach:

- "location\_map.pdf"

- "leak\_protocol.pdf"

escalate\_after: "2 hours"

#### Role-Based Task Routing

Emails are routed based on roles to ensure workflow continuity during personnel changes. For example, maintenance emails are routed to maintenance@organization.com, accessible by all team members.

#### Dynamic and Static Attachment Management

Attachments can be:

* **Dynamic**: Appended based on detected keywords (e.g., attaching a leak protocol for "leak").
* **Static**: Included for fixed scenarios (e.g., always attaching a standard incident report).

#### Feedback Loop Integration

Technician or user replies can trigger additional actions, such as escalating unresolved tasks or marking them as completed.

### 6.3 Core Workflow Components

#### 1. Trigger

An incoming email starts the workflow.

* **Example**: Email sent to emergency@organization.com with the subject "Water leak in Room 101."

#### 2. Parsing

Email metadata (subject, sender, body) is extracted and analyzed.

* **Example Extraction**:

Subject: "Water leak in Room 101."

Keywords: ["water", "leak"]

#### 3. Routing

The rule engine applies conditions to determine the recipient.

* **Example Rule Match**: Keywords "leak" and "water" → Route to maintenance@organization.com.

#### 4. Action

Emails are routed with relevant attachments and predefined templates.

* **Example Attachments**: Map of Room 101, Leak resolution protocol.

#### 5. Escalation

Tasks not addressed within a predefined timeframe are escalated.

* **Example**:

escalate\_after: "2 hours"

escalate\_to: "supervisor@organization.com"

#### 6. Resolution

Task completion is logged, and a summary is sent to the original sender or relevant parties.

### 6.4 Common Workflows

#### Maintenance Request Workflow

1. Trigger: Email to emergency@organization.com reporting a maintenance issue.
2. Parsing: Keywords like "leak" are identified.
3. Routing: Rule engine routes the email to maintenance@organization.com.
4. Attachments: Location map and protocol document are included.
5. Technician Response: ETA is logged, and the task is updated.
6. Escalation: If unresolved in 2 hours, escalate to a supervisor.

#### Harassment Reporting Workflow

1. Trigger: Employee emails report@organization.com.
2. Anonymization: Orgo anonymizes the sender and strips identifying metadata.
3. Routing: Email is routed to HR and legal advisors.
4. Attachments: Include harassment policy and investigation protocol.
5. Logging: All actions are logged for compliance audits.

#### IT Support Workflow

1. Trigger: Employee emails it@organization.com about a system outage.
2. Parsing: Keywords like "outage" are identified.
3. Routing: Email is routed to the IT support team.
4. Attachments: Logs of affected systems are included.
5. Resolution: IT team replies with updates or escalates to the infrastructure team.

### 6.5 Escalation Rule Standardization

#### Clear Conditions and Actions

Conditions define escalation triggers, and actions specify the next steps.

* **Example**:

escalation:

condition: "task unresolved in 2 hours"

action:

escalate\_to: "supervisor@organization.com"

notify:

- "manager@organization.com"

#### Escalation Levels

1. **Primary**: Notify team lead.
2. **Secondary**: Notify department manager.
3. **Final**: Escalate to organizational leadership.

### 6.6 Workflow Testing and Validation

#### Unit Testing

Verify individual steps:

* Email parsing.
* Rule matching.
* Routing and escalation.

#### End-to-End Testing

Simulate complete workflows:

* Validate routing accuracy.
* Test attachment handling (dynamic and static).
* Confirm timely escalations.

#### Stress Testing

Simulate high email volumes to ensure scalability.

### 6.7 Deliverables

* **Workflow Templates**: Predefined YAML/JSON rules for common workflows.
* **Workflow Logs**: Sample logs demonstrating routing decisions and escalations.
* **Testing Results**: Validation reports for simulated workflows.

### Summary

This section modularizes workflow logic by incorporating YAML templates and ensures standardization in attachment management and escalation rules. These enhancements provide scalable and efficient task management tailored to diverse organizational needs.