

**v3.4-STK-005 – Promptflow, Context & Specialization Logic**

Document ID: v3.4-STK-005 | Status: Final | Version: v3.4

Date: 2025-03-22

Author: Take Back Your Data – Interaction Systems

Document Type: Public / Certification / Internal

**1. Purpose & Scope**

This document defines the structure and execution logic of promptflows, context propagation and specialization for MaxOneOpen deployments. It enables interaction designers and LLM engineers to build adaptive, cost-efficient, and explainable response behavior.

**2. Promptflow Architecture**

- Prompts are structured into logical blocks:
  - System Primer
  - Role Alignment
  - Task Focus
  - Constraint Layer
- Each section is modular, controlled by context-aware handlers
- Promptflows are constructed dynamically per interaction type

Note: Interface structure and dynamic flow mapping logic are defined in Fork Blueprint §4.5.

**3. Context Layer Logic**

- Context is passed via structured metadata + soft memory
- System distinguishes between:
  - Volatile Context (per prompt)
  - Persistent Local Context (session-bound)
  - External Context Hints (from surrounding systems)
- Context policies define how memory is retained or dropped

Note: Context policy APIs and memory persistence rules are detailed in certified implementation schema.

**4. Specialization Models**

Type	Target	Mechanism
Prompt-based	Lightweight tuning	Preload system layer

Twin-selected	Task-specific inference	Activate matching twin
Role-routed	Persona/security routing	ControlAPI selector
Context-anchored	Behavior shaping	ContextPolicy + Mask Injection

Note: Specialization logic and routing schema definitions are provided in Fork Blueprint §4.6.

## 5. Optimization & Fallback Logic

- Reuse logic: If prompt+context pattern matches recent flow, reuse twin state
- Abort logic: Promptflow halts if constraint layer rejects unsafe combination
- Compression logic: Low-priority layers are token-minimized or dropped
- Fallback: If no specialization match, revert to default twin inference

Note: Optimization hooks and fallback transitions are maintained in fork-verified routing extensions.

## 6. Certification Relevance

Forks or systems claiming MaxOneOpen compatibility must implement or simulate promptflow logic with specialization and context-awareness. Prompt integrity, constraint enforcement and role-based routing are mandatory for certification.