### 

**Mastercard PrePaid Management Services (MPMS)**

**FIS Payments One Card Processing Platform (P1C)**

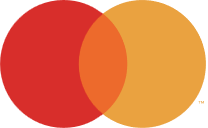
**Solution Architecture Specification**

**Solution Definition – Hardware, Software, Configuration and Connectivity**

*US Data Centres: Little Rock, AR and Brown Deer, W, Chandler, AZ*

*US Cloud Environments: Azure US East, Azure US Central*

### 

### 

### 

### 

### 

### **MPMS DOCUMENTATION OVERVIEW**

### **How the MPMS Application Works**

The MPMS application, based on the Payments One Card (P1C) platform, processes prepaid card transactions. This document provides a detailed explanation of the components, workflows, and database usage within the application.

#### **Key Components:**

1. **P1C Mainframe:** Central system processing all transactions.
2. **IST Switch:** Routes and processes transaction authorizations.
3. **FICO® Falcon:** Provides fraud detection and prevention.
4. **Data Express (DPR):** Manages file and report distribution.
5. **P1C Infinity Connect:** Handles API requests from external systems.
6. **P1C Service View:** Web portal for customer service and back-office functions.
7. **MoveIT:** Transfers files securely between systems.

#### **Process Overview:**

1. **Transaction Authorization:**
   * When a card transaction is initiated, it first reaches the Mastercard Information Processors (MIPs).
   * MIPs route the transaction to the IST Switch in the cloud.
   * The IST Switch then forwards the transaction to the P1C mainframe for authorization.
   * If the P1C mainframe is unavailable, the IST Switch can authorize the transaction using pre-set rules.
2. **Fraud Detection:**
   * As the transaction is processed, it is checked in real-time by FICO® Falcon.
   * Falcon scores the transaction for potential fraud and returns a recommendation to either approve or decline the transaction.
   * This ensures fraudulent transactions are detected and prevented.
3. **API Requests:**
   * External applications can interact with the P1C system through APIs managed by P1C Infinity Connect.
   * API requests are processed and, if necessary, cached locally to ensure high availability.
   * P1C Infinity Connect can handle requests even if the main P1C system is temporarily down.
4. **Customer Service:**
   * The P1C Service View provides a web portal for customer service representatives to manage cardholder accounts.
   * Representatives can perform actions such as checking account balances, viewing transaction history, and updating cardholder information.
5. **File and Report Distribution:**
   * Data Express (DPR) is used to manage the distribution of files and reports.
   * Nightly batch processes generate these files, which are then transmitted to the relevant parties.
6. **Data Transfer:**
   * MoveIT is used for secure file transfers between P1C and other internal systems.
   * It handles tasks like forwarding embossing files to card production facilities and sending data extracts to Watch List Reporter (WLR) for compliance checks.

#### **High Availability and Disaster Recovery:**

* **High Availability:** Components are distributed across multiple data centers and cloud regions.
* **Redundancy:** IST Switch, P1C mainframe, and P1C Infinity Connect have redundant setups.
* **Disaster Recovery:** Systems are backed up in different locations for quick recovery.

#### **Data Flows:**

1. **Authorization Flow:**
   * Mastercard Network → MIPs → IST Switch → P1C Mainframe (for authorization)
   * Stand-In Processing: IST Switch handles authorization if P1C Mainframe is down.
2. **Chargeback Data Flow:**
   * P1C Mainframe → Event Broker → CBK (Chargeback system)
   * CBK → P1C Infinity Connect (for updates)
3. **API Flow:**
   * External Application → P1C Infinity Connect → P1C Mainframe (for processing)
   * ICCO handles requests using cached data if the main system is down.
4. **Operational Data Flow:**
   * P1C Mainframe → Data Express → MPMS (for batch file delivery)
   * P1C Mainframe → Event Broker → MPMS (for real-time messages)

By leveraging these components and workflows, the MPMS application ensures reliable, secure, and efficient processing of prepaid card transactions, with robust fraud detection and high availability.

### **Database in the MPMS Application**

#### **Overview of Database Usage:**

The MPMS application uses IBM's DB2 database to store and manage cardholder data, transaction records, fraud detection logs, and operational metrics.

#### **Key Points about the Database:**

1. **DB2 Subsystems:**
   * **Production DB2 Subsystem:** Named DWP0.
   * **Non-Production DB2 Subsystem:** Named DWN0.
2. **Database Segregation:**
   * Data is segregated by schema within the DB2 subsystems.
3. **Data Set Naming Conventions:**
   * Specific naming conventions for organizing and managing data.
4. **Storage Considerations:**
   * Anticipates additional storage needs and meets PCI encryption requirements.

#### **Important Tables and Their Functions:**

1. **Cardholder Information Table:** Stores personal information, card status, and account balance.
2. **Transaction Records Table:** Logs transaction details.
3. **Fraud Detection Logs Table:** Records fraud check results.
4. **API Request Logs Table:** Tracks API requests and responses.
5. **Operational Metrics Table:** Collects data on system performance.

#### **Database Instances and Table Counts:**

1. **Production Environment:**
   * **Instance:** DWP0
   * **Estimated Number of Tables:** 100-200.
2. **Non-Production Environment:**
   * **Instance:** DWN0
   * **Estimated Number of Tables:** Similar to production with additional tables for testing and development.

#### **Key Database Features:**

1. **High Availability:** Replication across data centers, using Data Guard for real-time replication.
2. **Performance Optimization:** Use of Exadata slices for high performance.
3. **Security:** Pervasive encryption and role-based access controls (RACF).
4. **Data Integrity:** Regular backups, snapshots, and transaction logs.
5. **Scalability:** Designed to handle growing volumes of transactions and data.

### **Diagrams Explanation**

#### **1. High-Level Architecture Diagram**

**Explanation:** Shows the structure of the system, highlighting key components and their interactions.

* **Components:** P1C Mainframe, IST Switch, FICO® Falcon, P1C Infinity Connect.
* **Connectivity:** Secure connections, redundant setups for high availability.

**How It Works:**

* Transactions flow through the IST Switch to the P1C mainframe.
* Fraud checks are performed by FICO® Falcon.
* API requests are handled by P1C Infinity Connect with cached data if needed.
* Data is replicated across multiple locations.

#### **2. Authorization Flow Diagram**

**Explanation:** Illustrates the path a transaction takes from initiation to authorization.

* **Flow Steps:** Mastercard Network → MIPs → IST Switch → P1C Mainframe → FICO® Falcon.
* **Stand-In Processing:** IST Switch handles authorization if P1C Mainframe is unavailable.

**How It Works:**

* Transactions are routed from the Mastercard network to the IST Switch via MIPs.
* The IST Switch sends the transaction to the P1C mainframe for authorization.
* Fraud detection is performed by FICO® Falcon.
* If P1C Mainframe is down, the IST Switch uses pre-set rules for authorization.

#### **3. API Flow Diagram**

**Explanation:** Shows how API requests from external systems are processed.

* **Flow Steps:** External Systems → P1C Infinity Connect → ICCO → P1C Mainframe.
* **Cached Data:** ICCO handles requests using cached data if the main system is down.

**How It Works:**

* API requests are sent to P1C Infinity Connect.
* Requests are distributed and handled using cached data if possible.
* Requests needing mainframe processing are forwarded to the P1C mainframe.
* Responses are sent back through the same path.

#### **4. Chargeback Data Flow Diagram**

**Explanation:** Details how chargeback transactions are processed and communicated.

* **Flow Steps:** P1C Mainframe → Event Broker → CBK → P1C Infinity Connect.

**How It Works:**

* Chargeback data is processed by the P1C mainframe.
* Data is routed to the CBK system via the Event Broker.
* CBK processes chargebacks and updates status via API calls through P1C Infinity Connect.

#### **5. Operational Data Flow Diagram**

**Explanation:** Shows how operational data is transferred between P1C and MPMS.

* **Batch File Delivery:** P1C Mainframe → Data Express → MPMS.
* **Real-Time Messages:** P1C Mainframe → Event Broker → MPMS.

**How It Works:**

* Batch files are generated by the P1C mainframe and sent to Data Express, then transferred to MPMS.
* Real-time messages are generated by the P1C mainframe, sent to the Event Broker, and routed to MPMS.