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:**
   - **Mastercard Network:** Transactions originate and are routed to the IST Switch via Mastercard Information Processors (MIPs).
   - **IST Switch:** Forwards transactions to the P1C mainframe for authorization.
   - **P1C Mainframe:** Checks cardholder details and approves or declines transactions.
   - **Stand-In Processing:** If the P1C mainframe is unavailable, IST Switch authorizes transactions using pre-set rules.
2. **Fraud Detection:**
   - **FICO® Falcon:** Scores transactions for potential fraud and returns recommendations to approve or decline.
3. **API Requests:**
   - **P1C Infinity Connect:** Manages API requests, using cached data when the main system is down.
4. **Customer Service:**
   - **P1C Service View:** Allows customer service representatives to manage cardholder accounts, check balances, view transaction history, and update information.
5. **File and Report Distribution:**
   - **Data Express (DPR):** Manages nightly batch file distribution to relevant parties.
6. **Data Transfer:**
   - **MoveIT:** Secures file transfers between P1C and other systems, such as forwarding embossing files and sending data extracts 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 Involvement 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.

Functionality of WLR

1. **Data Input:**
    - WLR receives data inputs from various sources, including the MoveIT file transfer hub. This includes extracts of the full database to ensure comprehensive watch list processing.
2. **Screening:**
    - WLR screens the incoming data against various watch lists. These lists could include politically exposed persons (PEPs), sanctioned entities, and other high-risk individuals or organizations.
3. **Real-time Monitoring:**
    - WLR operates in real-time, meaning it continuously screens transactions and updates to ensure that any potentially risky entities are flagged immediately.
4. **Compliance Reporting:**
    - Once WLR identifies a match, it generates a report for compliance officers to review. This helps in ensuring that MPMS meets all regulatory requirements and avoids penalties.
5. **Integration with Other Systems:**
    - WLR integrates with other MPMS components such as P1C (Payments One Card) and Falcon (Fraud Detection and Interdiction). This integration ensures that flagged transactions can be quickly assessed and appropriate actions taken.

How It Works

1. **Data Collection:**
    - WLR collects data from P1C, which includes transaction details and account information. It also receives periodic database extracts via MoveIT.
2. **Watch List Matching:**
    - The collected data is matched against multiple watch lists using sophisticated algorithms. These watch lists are frequently updated to include the latest information on PEPs and sanctioned entities.
3. **Flagging Suspicious Entities:**
    - When WLR identifies a match, it flags the transaction or account for further investigation. This flagging helps in quickly isolating potentially risky activities.
4. **Generating Reports:**
    - WLR generates detailed reports on flagged transactions, which are then reviewed by compliance officers. These reports include all relevant information to help in decision-making.
5. **Actionable Insights:**

    - Based on the reports, compliance officers can take necessary actions such as freezing accounts, reporting to authorities, or further monitoring the flagged entities.

IST Switch in MPMS

Introduction: The IST Switch is a critical component in the Mastercard Payment Management System (MPMS) that handles the routing of transaction requests to the appropriate destination for authorization. It ensures that transaction data is efficiently processed, enabling real-time payment processing and communication between various parts of the system.

Key Functions of the IST Switch:

1. **Routing Transactions:**
    - Routes transaction requests from point-of-sale (POS) terminals, ATMs, and online gateways to the correct destination, such as the P1C Mainframe for authorization.
2. **Load Balancing:**
    - Distributes transaction requests evenly across multiple processing units to ensure optimal performance and prevent overloads.
3. **Protocol Conversion:**
    - Converts transaction requests into the appropriate protocol or format required by the destination system.
4. **Security and Compliance:**
    - Ensures secure transmission of transaction data, complying with industry standards and regulations.

Operational Process of the IST Switch:

1. **Transaction Initiation:**
    - A transaction begins at a POS terminal, ATM, or online payment gateway.
    - The transaction details, including card information, transaction amount, and merchant details, are sent to the IST Switch.
2. **Receiving Transaction Data:**
    - The IST Switch receives the transaction data and logs it for processing.
    - It performs initial validation checks to ensure the data is complete and in the correct format.
3. **Routing Decision:**
    - Based on predefined rules and the type of transaction, the IST Switch determines the appropriate destination for authorization (e.g., P1C Mainframe).
    - It uses routing tables and algorithms to make this decision efficiently.
4. **Protocol Conversion:**

    - If the destination system requires the data in a different protocol or format, the IST Switch converts the transaction data accordingly.
    - This ensures compatibility and seamless communication between systems.
5. **Sending for Authorization:**
    - The transaction data is sent to the designated authorization system (e.g., P1C Mainframe).
    - The IST Switch waits for a response from the authorization system.
6. **Receiving Authorization Response:**
    - The authorization system processes the transaction, verifying the card details, checking the available balance, and performing other necessary validations.
    - It sends an authorization response (approved or declined) back to the IST Switch.
7. **Forwarding Response:**
    - The IST Switch receives the authorization response and logs it.
    - It forwards the response back to the originating terminal (POS, ATM, or online gateway).
8. **Transaction Completion:**
    - The transaction is completed based on the authorization response.
    - If approved, the payment is processed, and the customer is notified.
    - If declined, the customer is informed, and the transaction is aborted.

Security Measures:

- The IST Switch employs various security measures to protect transaction data during transmission.
- It uses encryption, secure communication channels, and compliance with industry standards (e.g., PCI DSS) to ensure data integrity and confidentiality.

Benefits of the IST Switch:

- **Efficiency:** Ensures fast and efficient routing of transactions, minimizing processing time.
- **Scalability:** Can handle high volumes of transactions by distributing the load across multiple processing units.
- **Flexibility:** Supports various protocols and formats, enabling integration with different systems.
- **Reliability:** Provides a robust and reliable mechanism for transaction routing and authorization.

How ICCO(Infinity Connect Caching Service) Works:

1. **Chargeback Initiation:**
    - When a cardholder disputes a transaction, the issuer initiates a chargeback request.

- The chargeback request is entered into the system and transmitted to the acquirer.

2. **Document Collection:**
   - Both the issuer and acquirer upload necessary documentation to ICCO.
   - This includes all relevant evidence to support their position in the dispute.
3. **Communication and Correspondence:**
   - ICCO facilitates secure communication between the issuer, acquirer, and merchant.
   - Messages, documents, and updates are exchanged through the ICCO platform.
4. **Case Review and Decision:**
   - The acquirer reviews the chargeback request and corresponding documentation.
   - If the acquirer disagrees with the chargeback, they can provide counter-evidence through ICCO.
5. **Resolution and Updates:**
   - The chargeback case is reviewed by the card network, which makes a final decision based on the provided evidence.
   - ICCO updates the status of the chargeback case and informs all parties of the decision.
6. **Compliance and Record-Keeping:**
   - ICCO ensures that all actions taken during the chargeback process comply with regulations and network rules.
   - All communications and documents are stored securely for future reference and reporting.

Chargeback Data Flow Process:

1. **Chargeback Initiation:**
   - A chargeback process is initiated when a cardholder disputes a transaction.
   - The cardholder contacts their bank (issuer) to report the disputed transaction.
   - The issuer creates a chargeback request and sends it to the acquirer (merchant's bank).
2. **Data Entry and Transmission:**
   - The acquirer receives the chargeback request and enters it into the system.
   - The chargeback request is transmitted to the P1C Mainframe for processing.
3. **Processing in P1C Mainframe:**

   - The P1C Mainframe receives the chargeback request and matches it with the original transaction data.
   - It verifies the details and ensures the chargeback request is valid.
   - The mainframe updates the chargeback status and logs the information.
4. **Forwarding to CBK System:**
   - The chargeback data is forwarded to the CBK System for detailed processing.
   - The CBK System manages the chargeback cases, including the communication with the card networks (e.g., Mastercard) and the merchant.
5. **Communication with Card Networks:**
   - The CBK System communicates with the card networks to validate the chargeback request.
   - It exchanges data with the card networks to ensure compliance with their rules and regulations.
   - The card network may approve or reject the chargeback request based on their evaluation.
6. **Resolution and Updates:**
   - The CBK System processes the response from the card network and updates the chargeback status.
   - It coordinates with the acquirer and issuer to resolve the chargeback case.
   - If the chargeback is approved, the merchant account is debited, and the cardholder is credited.
   - The final resolution is logged in the system.
7. **Data Distribution via Data Express (DPR):**
   - The updated chargeback data is distributed to relevant systems and stakeholders.
   - Data Express (DPR) handles the nightly batch file distribution, ensuring all systems have the latest information.
8. **Secure Data Transfer with MoveIT:**
   - MoveIT ensures that the chargeback data is securely transferred between the P1C Mainframe, CBK System, and other systems.
   - It uses secure communication channels and encryption to protect the data during transfer.

High-Level Overview of Chargeback Data Flow:

- Cardholder disputes a transaction.
- Issuer creates a chargeback request and sends it to the acquirer.
- Acquirer enters the chargeback request into the system and transmits it to the P1C Mainframe.
- P1C Mainframe processes the chargeback request, verifies details, and forwards it to the CBK System.
- CBK System manages the chargeback case, communicates with card networks, and updates the status.
- Data Express (DPR) distributes updated chargeback data to relevant systems.
- MoveIT ensures secure data transfer between systems.

Components Involved in Caching:

1. **P1C Infinity Connect:**
    - This component is responsible for handling API requests from external systems.
    - It uses caching to manage and respond to these requests quickly, especially when the main system (P1C mainframe) is down or experiencing high load.

How Caching Works:

1. **Data Storage in Cache:**
    - Frequently accessed data, such as cardholder details, transaction history, and account balances, are stored in the cache.
    - This data is typically retrieved from the main database (DB2) and stored in memory for quick access.
2. **Handling API Requests:**
    - When an API request is received, P1C Infinity Connect first checks the cache for the required data.
    - If the data is available in the cache (cache hit), it is returned immediately, reducing the need for a database query and thus speeding up the response time.
    - If the data is not found in the cache (cache miss), P1C Infinity Connect retrieves it from the main database, processes the request, and updates the cache with the new data for future requests.
3. **Cache Refresh and Expiry:**
    - Cached data is periodically refreshed to ensure it remains up-to-date. This process involves re-fetching the latest data from the main database and updating the cache.
    - Cache expiry policies are implemented to automatically remove outdated data. This ensures that the cache only contains relevant and accurate information.

Benefits of Caching in MPMS:

1. **Improved Performance:**
    - By storing frequently accessed data in memory, the system can respond to API requests much faster than if it had to query the database each time.
    - This significantly reduces the load on the main database and improves overall system performance.
2. **High Availability:**
    - In scenarios where the main system (P1C mainframe) is down or slow, caching allows the system to continue serving API requests using the cached data.

    - This ensures that critical services remain available to users even during system outages or maintenance periods.
3. **Reduced Latency:**
    - Cached data can be retrieved much faster than querying the main database, resulting in lower latency for end-users and a better user experience.

Use Cases for Caching in MPMS:

1. **Cardholder Data:**
    - Information such as cardholder name, account balance, and recent transactions are cached to quickly serve account inquiries and transaction validations.
2. **Transaction History:**
    - Recent transaction records are cached to provide quick access for customer service representatives and automated fraud detection checks.
3. **API Responses:**
    - Responses to common API requests are cached to reduce the processing time and improve the speed of subsequent requests.

Challenges and Considerations:

1. **Data Consistency:**
    - Ensuring that the cached data remains consistent with the main database is crucial. This involves implementing effective cache invalidation and refresh strategies.
2. **Cache Size and Management:**
    - Determining the optimal size of the cache and managing it efficiently is essential to balance performance improvements and resource utilization.
3. **Security:**
    - Caching sensitive data requires robust security measures to protect against unauthorized access and data breaches.

MPMS API Gateway

External Connections:

- **APIGW1** - IP: 216.119.217.240
- **APIGW2** - IP: 216.119.209.240

How the API Gateway Works:

1. **Request Reception:**
   - External clients send API requests to the MPMS API Gateway using the provided public IP addresses (APIGW1 or APIGW2).
   - The requests typically arrive over HTTPS, ensuring data security during transmission.
2. **Security and Authentication:**
   - The API Gateway authenticates incoming requests. It uses mechanisms like mTLS (mutual TLS) to verify the identity of the client and ensure that only authorized entities can access the services.
   - Certificates are used for mutual authentication, where both client and server validate each other's identity before establishing a connection.
3. **Routing:**
   - Once authenticated, the API Gateway routes the requests to the appropriate internal MPMS service based on the API endpoint being accessed.
   - The routing rules are configured within the API Gateway to ensure that requests are directed to the correct internal servers or services.
4. **Load Balancing:**
   - The API Gateway balances the load across multiple instances of internal services to ensure high availability and reliability. This helps in managing traffic efficiently and prevents any single server from being overwhelmed.
5. **Data Processing:**
   - The internal MPMS services process the requests and perform the necessary business logic. This may involve accessing databases, performing transactions, or communicating with other internal systems.
6. **Response Handling:**
   - After processing the request, the internal services send the response back to the API Gateway.
   - The API Gateway then forwards the response back to the external client over the secure connection.
7. **Logging and Monitoring:**
   - The API Gateway logs all incoming and outgoing traffic. This helps in monitoring usage, detecting anomalies, and troubleshooting issues.
   - Metrics such as request rates, response times, and error rates are monitored to ensure the smooth functioning of the services.

Connectivity and Flow:

- **External Clients:** Connect to the MPMS API Gateway using the public IPs (APIGW1 or APIGW2). These connections are secured using TLS.
- **API Gateway:** Acts as the intermediary, handling authentication, routing, and load balancing.
- **Internal MPMS Services:** Receive requests from the API Gateway, process them, and send back the responses. These services are hosted within a secured internal network and are not directly accessible from the outside.