Power BI Implementation Plan for Alta's Unified Data Layer

This document outlines a comprehensive Power BI implementation plan for Alta's unified data layer, which integrates data from FieldRoutes and Dialpad. The plan focuses on structuring data for effective reporting, establishing relationships that support the unified data model, configuring hourly refresh settings, and recommending visualizations that provide valuable insights for customer and employee data.

Table of Contents

Dataset Design

Tables and Data Sources

The Power BI dataset will be structured to align with Alta's unified data model, which centers around customers and employees as primary entities. The dataset will include the following tables:

Fact Tables

Table Name	Source	Description	Key Fields
ServiceAppointments	Gold Layer - service_appointments.parquet	Contains all service appointment data	appointmentId, customerId, employeeId, scheduledStart, scheduledEnd, actualStart, actualEnd, status, type, amount
CommunicationRecords	Gold Layer - communication_records.parquet	Contains all communication records (calls, SMS, emails)	communicationId, customerId, employeeId, timestamp, type, direction, duration, status
CustomerInteractions	Gold Layer - customer_interactions.parquet	Contains all customer interaction data	interactionId, customerId, employeeId, timestamp, type, channel, outcome

Dimension Tables

Table Name	Source	Description	Key Fields
Customers	Gold Layer - customers.parquet	Master customer data	customerId, name, email, phone, status, address, createdDate, modifiedDate, type, tags
Employees	Gold Layer - employees.parquet	Master employee data	employeeId, name, email, phone, role, department, status, hireDate, skills

Table Name	Source	Description	Key Fields
Date	Generated in Power BI	Date dimension for time intelligence	dateKey, date, day, month, quarter, year, isWeekday, fiscalPeriod
Time	Generated in Power BI	Time dimension for hourly analysis	timeKey, hour, minute, hourOfDay, amPm, timeOfDayCategory
Geography	Derived from customer addresses	Geographical dimension	geoKey, city, state, zip, country, region

Measures and Calculated Columns

Key Calculated Columns Customers Table:

```
CustomerFullAddress = Customers[street] & ", " & Customers[city] & ", " & Customers[state] &

CustomerAgeInDays = DATEDIFF(Customers[createdDate], TODAY(), DAY)

CustomerCategory =
SWITCH(
    TRUE(),
    Customers[CustomerAgeInDays] <= 30, "New",
    Customers[CustomerAgeInDays] <= 365, "Established",
    "Long-term"
)
```

ServiceAppointments Table:

```
AppointmentDuration = DATEDIFF(ServiceAppointments[actualEnd], ServiceAppointments[actualSta

IsOnTime =
IF(
    ServiceAppointments[actualStart] <= ServiceAppointments[scheduledStart],
    "Yes",
    "No"
)

ServiceEfficiency =
DIVIDE(
    DATEDIFF(ServiceAppointments[scheduledEnd], ServiceAppointments[scheduledStart], MINUTE)
    ServiceAppointments[AppointmentDuration],
    1
```

```
)
```

CommunicationRecords Table:

```
CommunicationDayOfWeek = FORMAT(CommunicationRecords[timestamp], "dddd")

CommunicationHourOfDay = HOUR(CommunicationRecords[timestamp])

CommunicationTimeCategory =
SWITCH(
    TRUE(),
    CommunicationRecords[CommunicationHourOfDay] < 9, "Early Morning",
    CommunicationRecords[CommunicationHourOfDay] < 12, "Morning",
    CommunicationRecords[CommunicationHourOfDay] < 17, "Afternoon",
    CommunicationRecords[CommunicationHourOfDay] < 20, "Evening",
    "Night"
)
```

Key Measures Customer Metrics:

```
Total Customers = COUNTROWS(Customers)

Active Customers = CALCULATE(COUNTROWS(Customers), Customers[status] = "Active")

Customer Retention Rate =
DIVIDE(
    CALCULATE(COUNTROWS(Customers), Customers[status] = "Active", Customers[CustomerAgeInDay
    CALCULATE(COUNTROWS(Customers), Customers[CustomerAgeInDays] > 365),
    0
)

Average Service Value =
DIVIDE(
    SUM(ServiceAppointments[amount]),
    DISTINCTCOUNT(ServiceAppointments[customerId]),
    0
)
```

Employee Metrics:

```
Total Employees = COUNTROWS(Employees)

Active Technicians = CALCULATE(COUNTROWS(Employees), Employees[role] = "Technician", Employe

Average Appointments Per Technician =
DIVIDE(
    COUNTROWS(ServiceAppointments),
    [Active Technicians],
```

```
    0
)

Average Service Duration =
AVERAGEX(
    ServiceAppointments,
    ServiceAppointments[AppointmentDuration]
)
```

Communication Metrics:

```
Total Communications = COUNTROWS(CommunicationRecords)

Inbound Communications = CALCULATE(COUNTROWS(CommunicationRecords), CommunicationRecords[dir

Outbound Communications = CALCULATE(COUNTROWS(CommunicationRecords), CommunicationRecords[di

Average Call Duration =
AVERAGEX(
    FILTER(CommunicationRecords, CommunicationRecords[type] = "Call"),
    CommunicationRecords[duration]
)

Response Rate =
DIVIDE(
    CALCULATE(COUNTROWS(CommunicationRecords), CommunicationRecords[direction] = "Outbound")
    CALCULATE(COUNTROWS(CommunicationRecords), CommunicationRecords[direction] = "Inbound"),
    0
)
```

Data Modeling Best Practices

1. **Naming Conventions:**
   - Use PascalCase for table names (e.g., ServiceAppointments)
   - Use camelCase for column names (e.g., customerId)
   - Use Title Case for measure names (e.g., Average Service Duration)
2. **Performance Optimization:**
   - Implement aggregation tables for large fact tables
   - Use integers for key columns where possible
   - Avoid unnecessary calculated columns; use measures instead
   - Implement appropriate data types for each column
3. **Usability Considerations:**
   - Organize measures into display folders by functional area
   - Hide unnecessary columns from report view
   - Create hierarchies for common drill-down paths
   - Add descriptions to tables, columns, and measures

Relationship Modeling

Star Schema Design

The Power BI data model will follow a star schema design, with fact tables at the center connected to dimension tables. This approach optimizes query performance and simplifies the user experience when creating reports.

Relationship Diagram

```
erDiagram
    CUSTOMERS ||--o{ SERVICE_APPOINTMENTS : "has"
    CUSTOMERS ||--o{ COMMUNICATION_RECORDS : "participates in"
    CUSTOMERS ||--o{ CUSTOMER_INTERACTIONS : "has"
    EMPLOYEES ||--o{ SERVICE_APPOINTMENTS : "performs"
    EMPLOYEES ||--o{ COMMUNICATION_RECORDS : "handles"
    EMPLOYEES ||--o{ CUSTOMER_INTERACTIONS : "conducts"
    DATE ||--o{ SERVICE_APPOINTMENTS : "scheduled on"
    DATE ||--o{ COMMUNICATION_RECORDS : "occurred on"
    DATE ||--o{ CUSTOMER_INTERACTIONS : "happened on"
    TIME ||--o{ SERVICE_APPOINTMENTS : "scheduled at"
    TIME ||--o{ COMMUNICATION_RECORDS : "occurred at"
    GEOGRAPHY ||--o{ CUSTOMERS : "located in"
```

Active vs. Inactive Relationships

Most relationships in the model will be active, allowing for automatic filtering across tables. However, some specific scenarios require inactive relationships with explicit activation in DAX measures:

1. **Multiple Date References:** For tables with multiple date fields (e.g., scheduledStart and actualStart in ServiceAppointments), create active relationships to the primary date reference and inactive relationships to secondary date references.
2. **Role-Playing Dimensions:** The Date dimension will be role-playing (e.g., ScheduleDate, ActualDate) with inactive relationships that are activated in specific measures.

Example of a measure using an inactive relationship:

```
Appointments by Actual Date =
CALCULATE(
    COUNTROWS(ServiceAppointments),
    USERELATIONSHIP(ServiceAppointments[actualStartDate], Date[DateKey])
)
```

Refresh Settings

Hourly Refresh Configuration

To support Alta's requirement for hourly data refreshes, the following configuration will be implemented:

1. **Power BI Premium Capacity:** Utilize Premium capacity to enable more frequent refreshes (up to 48 times per day).
2. **Refresh Schedule:** Configure scheduled refreshes to run hourly, starting at the top of each hour.
3. **Refresh Monitoring:** Implement refresh monitoring using Power BI REST API to track refresh history and detect failures.
4. **Notification System:** Set up email notifications for refresh failures to alert administrators.

Incremental Refresh Strategy

To optimize refresh performance and reduce load on source systems, implement incremental refresh policies:

```
// Example incremental refresh policy configuration
RangeStart = DateTime.Date(DateTimeZone.UtcNow()) - Duration.Days(7)
RangeEnd = DateTime.Date(DateTimeZone.UtcNow())
IsInRange = [timestamp] >= RangeStart && [timestamp] < RangeEnd
```

Configure incremental refresh with the following parameters: - Store rows in the last 7 days - Refresh rows in the last 1 day - Detect data changes based on the modified timestamp

Gateway Configuration

1. **On-Premises Data Gateway:** Deploy in high-availability mode with multiple gateway instances.
2. **Gateway Monitoring:** Implement monitoring of gateway performance and availability.
3. **Connection Settings:**
   - Configure timeout settings to accommodate larger data volumes
   - Implement query folding where possible to optimize performance
   - Configure appropriate privacy levels for data sources

Report Types and Visualizations

Customer Analytics Dashboard

Purpose: Provide comprehensive insights into customer behavior, service history, and communication patterns.

Key Visualizations: 1. **Customer Overview:** - Card visuals showing total customers, active customers, and customer retention rate - Donut chart showing customer distribution by type - Line chart showing customer growth over time

2. **Service Analysis:**
   - Bar chart of service appointments by type
   - Heat map of service appointments by day and hour
   - Scatter plot of service duration vs. service value
3. **Communication Analysis:**
   - Line chart of communication volume by channel (call, SMS, email)
   - Bar chart of average response time by communication type
   - Funnel visual showing communication-to-appointment conversion

Mockup:

```
graph TD
    subgraph "Customer Analytics Dashboard"
        subgraph "Customer Overview"
            A[Total Customers<br>Card Visual]
            B[Active Customers<br>Card Visual]
            C[Retention Rate<br>Card Visual]
            D[Customer Types<br>Donut Chart]
            E[Customer Growth<br>Line Chart]
        end

        subgraph "Service Analysis"
            F[Service by Type<br>Bar Chart]
            G[Service Heatmap<br>Matrix Visual]
            H[Duration vs Value<br>Scatter Plot]
            I[Service Trends<br>Line Chart]
        end

        subgraph "Communication Analysis"
            J[Communication Volume<br>Line Chart]
            K[Response Time<br>Bar Chart]
            L[Communication Funnel<br>Funnel Visual]
            M[Communication by Hour<br>Column Chart]
        end
    end
```

Employee Performance Dashboard

Purpose: Track and analyze employee performance metrics, focusing on technicians and customer service representatives.

Key Visualizations: 1. **Employee Overview:** - Card visuals showing total employees, active technicians, and average appointments per technician - Donut chart showing employee distribution by role - Bar chart showing employees by department

2. **Technician Performance:**
   - Gauge charts showing on-time arrival percentage
   - Bar chart of average service duration by technician
   - Scatter plot of service quality vs. efficiency
3. **Communication Performance:**
   - Line chart of call volume by employee
   - Bar chart of average call duration
   - Heat map of customer satisfaction by employee

Mockup:

```
graph TD
    subgraph "Employee Performance Dashboard"
        subgraph "Employee Overview"
            A[Total Employees<br>Card Visual]
            B[Active Technicians<br>Card Visual]
            C[Avg Appointments<br>Card Visual]
            D[Roles Distribution<br>Donut Chart]
            E[Departments<br>Bar Chart]
        end

        subgraph "Technician Performance"
            F[On-Time Percentage<br>Gauge Chart]
            G[Service Duration<br>Bar Chart]
            H[Quality vs Efficiency<br>Scatter Plot]
            I[Appointments Completed<br>Column Chart]
        end

        subgraph "Communication Performance"
            J[Call Volume<br>Line Chart]
            K[Call Duration<br>Bar Chart]
            L[Customer Satisfaction<br>Heat Map]
            M[Response Rate<br>KPI Visual]
        end
    end
```

Operational Metrics Dashboard

Purpose: Monitor day-to-day operations, identify bottlenecks, and optimize resource allocation.

Key Visualizations: 1. **Daily Operations:** - Line chart of appointment volume by day - Stacked column chart of appointment status by day - Gauge chart showing resource utilization

2. **Efficiency Metrics:**
   - Waterfall chart showing time allocation
   - Line chart of average service duration trend
   - Bar chart of service completion rate by type
3. **Geographic Analysis:**
   - Map visual showing appointment density by region
   - Column chart of travel time by region
   - Heat map of service demand by zip code

Mockup:

```
graph TD
    subgraph "Operational Metrics Dashboard"
        subgraph "Daily Operations"
            A[Appointment Volume<br>Line Chart]
            B[Appointment Status<br>Stacked Column]
            C[Resource Utilization<br>Gauge Chart]
            D[Upcoming Appointments<br>Table Visual]
        end

        subgraph "Efficiency Metrics"
            E[Time Allocation<br>Waterfall Chart]
            F[Service Duration<br>Line Chart]
            G[Completion Rate<br>Bar Chart]
            H[Efficiency KPIs<br>Multi-row Card]
        end

        subgraph "Geographic Analysis"
            I[Appointment Density<br>Map Visual]
            J[Travel Time<br>Column Chart]
            K[Service Demand<br>Heat Map]
            L[Region Filter<br>Slicer]
        end
    end
```

Executive Summary Dashboard

Purpose: Provide high-level insights for executive decision-making, focusing on key performance indicators and trends.

Key Visualizations: 1. **Business Overview:** - Scorecard showing key metrics vs. targets - Small multiples of trend lines for critical KPIs - Decomposition tree for revenue analysis

2. **Customer Insights:**
   - Ribbon chart showing customer segment performance
   - Key influencers visual for customer retention
   - Smart narrative summarizing customer trends
3. **Operational Excellence:**
   - KPI indicators for operational metrics
   - Forecast chart for service demand
   - Anomaly detection for operational disruptions

Mockup:

```
graph TD
    subgraph "Executive Summary Dashboard"
        subgraph "Business Overview"
            A[KPI Scorecard<br>Multi-row Card]
            B[Trend Lines<br>Small Multiples]
            C[Revenue Analysis<br>Decomposition Tree]
            D[YoY Comparison<br>Column Chart]
        end

        subgraph "Customer Insights"
            E[Segment Performance<br>Ribbon Chart]
            F[Retention Factors<br>Key Influencers]
            G[Customer Trends<br>Smart Narrative]
            H[Forecast<br>Line Chart]
        end

        subgraph "Operational Excellence"
            I[Operational KPIs<br>Indicator]
            J[Service Demand<br>Forecast Chart]
            K[Anomaly Detection<br>Line Chart]
            L[Resource Planning<br>What-if Parameter]
        end
    end
```

Security, Sharing, and User Access

Row-Level Security (RLS)

Implement row-level security to restrict data access based on user roles and responsibilities:

1. **Department-Based Security:**

   ```
   // Example RLS expression for department managers
   ```

```
[department] = USERNAME() || LOOKUPVALUE(SecurityTable[AllowedDepartments], SecurityTab
```

2. **Region-Based Security:**

```
// Example RLS expression for regional managers
[region] IN VALUES(SecurityTable[AllowedRegions]) || LOOKUPVALUE(SecurityTable[IsExecut
```

3. **Role-Based Security:**

```
// Example RLS expression for role-specific access
LOOKUPVALUE(SecurityTable[AllowedRoles], SecurityTable[Email], USERNAME()) = "Admin" ||
[role] IN SPLIT(LOOKUPVALUE(SecurityTable[ViewableRoles], SecurityTable[Email], USERNAM
```

Report Distribution

1. **App Workspace Structure:**
   - Create separate workspaces for different functional areas (Customer Operations, Field Service, Executive)
   - Organize reports within each workspace by purpose and audience
   - Implement consistent naming conventions for all reports
2. **Subscription and Alerts:**
   - Configure email subscriptions for regular report distribution
   - Set up data-driven alerts for critical metrics
   - Schedule report snapshots for key stakeholders
3. **Embedding Options:**
   - Embed key reports in internal portals or applications
   - Create mobile-optimized views for field staff
   - Implement secure embedding using Azure AD authentication

User Roles and Permissions

Role	Description	Access Level	Workspace Permissions
Executives	C-level and senior management	Full access to all reports with executive views	Admin access to Executive workspace, Viewer access to all other workspaces
Department Managers	Managers of specific departments	Access to department-specific data and aggregated company data	Member access to department workspace, Viewer access to Executive workspace
Field Supervisors	Supervisors of field technicians	Access to technician performance and service data	Member access to Field Service workspace

Role	Description	Access Level	Workspace Permissions
Customer Service Managers	Managers of customer service teams	Access to communication data and customer interactions	Member access to Customer Operations workspace
Analysts	Business and data analysts	Full access to all data for analysis purposes	Contributor access to all workspaces
General Staff	Regular employees	Limited access to relevant operational reports	Viewer access to specific reports

Power BI Deployment Options

Power BI Service vs. Premium

Based on Alta's requirements for hourly data refreshes and the complexity of the data model, we recommend implementing **Power BI Premium Per Capacity (P1)** for the following reasons:

1. **Refresh Frequency:**
   - Power BI Pro: Limited to 8 refreshes per day
   - Power BI Premium: Up to 48 refreshes per day, supporting Alta's hourly refresh requirement
2. **Model Size Limits:**
   - Power BI Pro: 1 GB model size limit
   - Power BI Premium P1: 25 GB model size limit, accommodating Alta's growing data volume
3. **Advanced Features:**
   - Incremental refresh for optimized data loading
   - Paginated reports for operational reporting needs
   - AI capabilities for advanced analytics
   - Dataflows for centralized data preparation
4. **Performance:**
   - Dedicated capacity for consistent performance
   - Larger memory allocation for complex calculations
   - Parallel refresh operations

Deployment Architecture

```
flowchart TD
    subgraph "Data Sources"
        A[Azure Synapse Analytics]
        B[Gold Layer Data]
```

```
end

subgraph "Power BI Premium Capacity"
    C[Dataflows]
    D[Datasets]
    E[Reports]
    F[Dashboards]
    G[Paginated Reports]
end

subgraph "Distribution Channels"
    H[Power BI Service]
    I[Power BI Mobile]
    J[Embedded Analytics]
    K[Email Subscriptions]
end

subgraph "Security & Governance"
    L[Azure AD]
    M[Row-Level Security]
    N[Workspace Roles]
    O[Usage Monitoring]
end

A --> B
B --> C
C --> D
D --> E
D --> G
E --> F

F --> H
F --> I
E --> J
F --> K
G --> K

L --> M
L --> N
H --> O
I --> O
J --> O
```

Capacity Planning

For Alta's implementation, we recommend starting with a P1 capacity with the following allocation:

Resource	Allocation	Notes
Memory	25 GB	Allocated primarily to datasets (15 GB), dataflows (5 GB), and paginated reports (5 GB)
CPU	4 cores	Distributed across dataset refresh, report rendering, and AI functions
Concurrent Refreshes	6	Sufficient for hourly refreshes of multiple datasets
Paginated Reports	Enabled	For operational reporting needs
AI Functions	Enabled	For advanced analytics capabilities

Estimated Cost: Approximately $5,000/month for P1 capacity

Scaling Strategy: - Monitor capacity metrics using the Power BI Premium Capacity Metrics app - Implement auto-scale during business hours if needed - Consider upgrading to P2 if consistent memory pressure is observed

Implementation Roadmap

Phase 1: Foundation (Weeks 1-4)

1. **Dataset Design and Development:**
   - Create the core data model with primary tables and relationships
   - Implement key measures and calculated columns
   - Configure incremental refresh policies
2. **Core Reports Development:**
   - Develop Customer Analytics Dashboard
   - Develop Employee Performance Dashboard
   - Create basic operational reports
3. **Security Implementation:**

- Configure workspace structure and permissions
- Implement row-level security rules
- Set up user groups in Azure AD

4. **Refresh Configuration:**
   - Set up on-premises data gateway
   - Configure hourly refresh schedule
   - Implement refresh monitoring

Phase 2: Advanced Features (Weeks 5-8)

1. **Enhanced Analytics:**
   - Implement advanced DAX measures
   - Create AI-powered visualizations
   - Develop what-if analysis capabilities
2. **Additional Dashboards:**
   - Develop Operational Metrics Dashboard
   - Create Executive Summary Dashboard
   - Implement mobile-optimized views
3. **Automation and Alerts:**
   - Set up data-driven alerts
   - Configure scheduled report distribution
   - Implement dashboard subscriptions
4. **Integration:**
   - Embed key reports in internal portals
   - Integrate with Microsoft Teams
   - Configure mobile app access

Phase 3: Optimization (Weeks 9-12)

1. **Performance Tuning:**
   - Optimize DAX measures
   - Implement aggregation tables
   - Fine-tune refresh policies
2. **User Adoption:**
   - Conduct user training sessions
   - Create self-service documentation
   - Gather and implement user feedback
3. **Governance:**
   - Implement dataset certification
   - Set up usage monitoring
   - Create governance documentation
4. **Future Planning:**
   - Evaluate capacity utilization
   - Plan for additional data sources
   - Develop roadmap for advanced features

Conclusion

This Power BI implementation plan provides a comprehensive framework for Alta to leverage their unified data layer for effective reporting and analysis. By following this plan, Alta will be able to:

1. Create a well-structured dataset that aligns with their unified data model
2. Establish relationships that support comprehensive analysis across customer and employee data
3. Configure refresh settings to meet the hourly data refresh requirement
4. Implement visualizations that provide valuable insights for decision-making
5. Secure and share reports with appropriate stakeholders
6. Deploy Power BI in a scalable and cost-effective manner

The phased implementation approach allows for incremental development and validation, ensuring that the solution meets Alta's requirements while providing opportunities for feedback and refinement throughout the process.