

# FHIR Patient Portal - Complete API Specifications

## Table of Contents

- 1. HL7 Parser Service
- 2. Patient Service
- 3. Observation Service
- 4. Appointment Service
- 5. User Auth Service
- 6. Role Management Service
- 7. Telemedicine Service
- 8. Notification Service
- 9. Analytics Service
- 10. Audit Logging Service
- 11. FHIR API Gateway
- 12. Kafka Communication Patterns

## HL7 Parser Service

Base URL: `/api/v1/hl7-parser`

### 1. HL7 Message Ingestion

Endpoint: `POST /ingest`

#### Request Headers:

Content-Type: application/hl7-v2  
Authorization: Bearer {jwt\_token}

#### Request Payload:

MSH|^~\&|SENDING\_APP|SENDING\_FACILITY|RECEIVING\_APP|RECEIVING\_FACILITY|20230901123045||ADT^AC  
EVN||20230901123045|||USER123  
PID|||MRN12345||Doe^John^M||19800115|M|||123 Main St^^City^ST^12345^USA||5551234567|  
PV1|||ICU^101^A||DOC123^Smith^Jane^MD||SUR|||A||DOC123|INS|12345|||||||||20230901123045

#### Response Payload:

json

```
{
  "messageId": "MSG001",
  "correlationId": "corr-uuid-123",
  "status": "processed",
  "timestamp": "2023-09-01T12:30:45Z",
  "fhirResources": [
    {
      "resourceType": "Patient",
      "id": "patient-12345",
      "identifier": [{"value": "MRN12345"}],
      "name": [{"family": "Doe", "given": ["John"]}]}
  ],
  "errors": []
}
```

### Validation Rules:

- MSH segment must be present
- Message Control ID must be unique
- Sending/Receiving applications must be registered
- HL7 version must be 2.3-2.8

### Kafka Events:

- Topic: `hl7.message.received`
- Topic: `fhir.resource.created`

## 2. Message Parsing Status

**Endpoint:** `GET /parse-status/{messageId}`

### Response Payload:

json

```
{
  "messageId": "MSG001",
  "status": "completed",
  "processedAt": "2023-09-01T12:30:45Z",
  "resourcesCreated": 3,
  "errors": []
}
```

### 3. Batch Processing

Endpoint: `POST /batch`

Request Payload:

```
json

{
  "batchId": "batch-001",
  "messages": [
    {
      "messageId": "MSG001",
      "content": "MSH|^~\\&|...",
      "priority": "normal"
    }
  ]
}
```

Response Payload:

```
json

{
  "batchId": "batch-001",
  "totalMessages": 10,
  "processed": 8,
  "failed": 2,
  "processingTime": "45.2s",
  "status": "partial_success"
}
```

---

### Patient Service

Base URL: `/api/v1/patients`

#### 4. Patient Registration

Endpoint: `POST /`

Request Headers:

```
Content-Type: application/fhir+json
Authorization: Bearer {jwt_token}
```

## Request Payload:

```
json

{
  "resourceType": "Patient",
  "identifier": [
    {
      "use": "usual",
      "type": {"coding": [{"system": "http://terminology.hl7.org/CodeSystem/v2-0203", "code": "MR"}]},
      "value": "MRN12345"
    }
  ],
  "name": [
    {
      "use": "official",
      "family": "Doe",
      "given": ["John", "Michael"]
    }
  ],
  "gender": "male",
  "birthDate": "1980-01-15",
  "address": [
    {
      "use": "home",
      "line": ["123 Main St"],
      "city": "Springfield",
      "state": "IL",
      "postalCode": "62701",
      "country": "US"
    }
  ],
  "telecom": [
    {
      "system": "phone",
      "value": "555-123-4567",
      "use": "mobile"
    },
    {
      "system": "email",
      "value": "john.doe@email.com"
    }
  ]
}
```

## Response Payload:

json

```
{
  "resourceType": "Patient",
  "id": "patient-uuid-123",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2023-09-01T12:30:45Z"
  },
  "identifier": [
    {
      "use": "usual",
      "type": {"coding": [{"system": "http://terminology.hl7.org/CodeSystem/v2-0203", "code": "MR"}]},
      "value": "MRN12345"
    }
  ],
  "name": [
    {
      "use": "official",
      "family": "Doe",
      "given": ["John", "Michael"]
    }
  ],
  "gender": "male",
  "birthDate": "1980-01-15"
}
```

### Validation Rules:

- At least one identifier required
- Valid email format if provided
- Phone number format validation
- Birth date cannot be in future
- Gender must be from ValueSet
- Required fields: name, gender, birthDate

### Kafka Events:

- Topic: `patient.registered`
- Topic: `audit.patient.created`

## 5. Patient Profile Update

**Endpoint:** `PUT /{patientId}`

**Request Payload:** (Same as registration with modifications)

**Validation Rules:**

- Patient must exist
- User must have permission to update
- Audit trail required for all changes

## 6. Patient Search

**Endpoint:** GET /search

**Query Parameters:**

```
?identifier=MRN12345
?name=John
?birthdate=1980-01-15
?phone=555-123-4567
?email=john.doe@email.com
?_fuzzy=true
```

**Response Payload:**

```
json

{
  "resourceType": "Bundle",
  "type": "searchset",
  "total": 1,
  "entry": [
    {
      "resource": {
        "resourceType": "Patient",
        "id": "patient-uuid-123",
        "name": [{"family": "Doe", "given": ["John"]}]}
    }
  ]
}
```

## 7. Patient Record Merge

**Endpoint:** POST /{sourceId}/merge/{targetId}

**Request Payload:**

json

```
{
  "reason": "Duplicate records identified",
  "mergeStrategy": "keep_latest",
  "fields": ["contact", "address"],
  "auditReason": "Data quality improvement"
}
```

### Response Payload:

json

```
{
  "status": "merged",
  "resultPatientId": "patient-uuid-123",
  "mergedFields": ["telecom", "address"],
  "auditId": "audit-uuid-456"
}
```

## 8. Data Export

**Endpoint:** `GET /{patientId}/export`

### Query Parameters:

?format=pdf|fhir|cda  
?includeSections=demographics,vitals,labs,medications

### Response Payload:

json

```
{
  "exportId": "export-uuid-789",
  "status": "completed",
  "downloadUrl": "/api/v1/exports/export-uuid-789/download",
  "format": "pdf",
  "size": "2.4MB",
  "expiresAt": "2023-09-08T12:30:45Z"
}
```

---

# Observation Service

Base URL: `/api/v1/observations`

## 9. Vital Signs Entry

Endpoint: `POST /`

Request Payload:

```
json
```



```
{
  "resourceType": "Observation",
  "status": "final",
  "category": [
    {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/observation-category",
          "code": "vital-signs"
        }
      ]
    }
  ],
  "code": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "85354-9",
        "display": "Blood pressure panel with all children optional"
      }
    ]
  },
  "subject": {
    "reference": "Patient/patient-uuid-123"
  },
  "effectiveDateTime": "2023-09-01T12:30:45Z",
  "component": [
    {
      "code": {
        "coding": [
          {
            "system": "http://loinc.org",
            "code": "8480-6",
            "display": "Systolic blood pressure"
          }
        ]
      },
      "valueQuantity": {
        "value": 120,
        "unit": "mmHg",
        "system": "http://unitsofmeasure.org",
        "code": "mm[Hg]"
      }
    }
  ]
}
```

```
]
}
```

### Response Payload:

```
json

{
  "resourceType": "Observation",
  "id": "obs-uuid-123",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2023-09-01T12:30:45Z"
  },
  "status": "final",
  "category": [
    {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/observation-category",
          "code": "vital-signs"
        }
      ]
    }
  ]
}
```

### Validation Rules:

- LOINC codes must be valid
- Units must be UCUM compliant
- Vital sign ranges validation
- Patient reference must exist
- Effective date cannot be future

## 10. Lab Results Integration

**Endpoint:** `POST /lab-results`

### Request Payload:

```
json
```

```

{
  "resourceType": "Bundle",
  "type": "transaction",
  "entry": [
    {
      "request": {
        "method": "POST",
        "url": "Observation"
      },
      "resource": {
        "resourceType": "Observation",
        "status": "final",
        "category": [
          {
            "coding": [
              {
                "system": "http://terminology.hl7.org/CodeSystem/observation-category",
                "code": "laboratory"
              }
            ]
          }
        ],
        "code": {
          "coding": [
            {
              "system": "http://loinc.org",
              "code": "2339-0",
              "display": "Glucose"
            }
          ]
        },
        "valueQuantity": {
          "value": 95,
          "unit": "mg/dL",
          "system": "http://unitsofmeasure.org"
        }
      }
    }
  ]
}

```

## 11. Trend Visualization Data

**Endpoint:** GET `/f{patientId}/trends`

**Query Parameters:**

?code=2339-0 // LOINC code for glucose

?category=laboratory

?period=P30D // Last 30 days

## Response Payload:

json

```
{
  "patientId": "patient-uuid-123",
  "observationType": "glucose",
  "unit": "mg/dL",
  "timeRange": {
    "start": "2023-08-01T00:00:00Z",
    "end": "2023-09-01T00:00:00Z"
  },
  "dataPoints": [
    {
      "timestamp": "2023-08-01T08:00:00Z",
      "value": 95,
      "status": "normal"
    },
    {
      "timestamp": "2023-08-15T08:00:00Z",
      "value": 110,
      "status": "high"
    }
  ],
  "referenceRanges": {
    "low": 70,
    "high": 100
  }
}
```

## 12. Clinical Alerts

**Endpoint:** `POST /alerts/configure`

### Request Payload:

json

```
{
  "patientId": "patient-uuid-123",
  "observationCode": "8480-6",
  "thresholds": {
    "critical": {
      "high": 180,
      "low": 90
    },
    "warning": {
      "high": 140,
      "low": 100
    }
  },
  "notificationChannels": ["email", "sms", "app"],
  "active": true
}
```

### Kafka Events:

- Topic: `observation.created`
- Topic: `alert.triggered`
- Topic: `trend.calculated`

---

## Appointment Service

Base URL: `/api/v1/appointments`

### 13. Appointment Booking

Endpoint: `POST /`

Request Payload:

json

```
{
  "resourceType": "Appointment",
  "status": "booked",
  "serviceCategory": [
    {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/service-category",
          "code": "17",
          "display": "General Practice"
        }
      ]
    }
  ],
  "appointmentType": {
    "coding": [
      {
        "system": "http://terminology.hl7.org/CodeSystem/v2-0276",
        "code": "ROUTINE"
      }
    ]
  },
  "start": "2023-09-15T09:00:00Z",
  "end": "2023-09-15T09:30:00Z",
  "participant": [
    {
      "actor": {
        "reference": "Patient/patient-uuid-123"
      },
      "required": "required",
      "status": "accepted"
    },
    {
      "actor": {
        "reference": "Practitioner/doc-uuid-456"
      },
      "required": "required",
      "status": "accepted"
    }
  ]
}
```

## Response Payload:

json

```
{
  "resourceType": "Appointment",
  "id": "appt-uuid-789",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2023-09-01T12:30:45Z"
  },
  "status": "booked",
  "start": "2023-09-15T09:00:00Z",
  "end": "2023-09-15T09:30:00Z",
  "confirmationCode": "CONF123456"
}
```

### Validation Rules:

- Start time must be in future
- End time must be after start time
- Practitioner must be available
- Patient cannot have overlapping appointments
- Business hours validation

## 14. Real-time Availability

**Endpoint:** `GET /availability`

### Query Parameters:

```
?practitioner=doc-uuid-456
?date=2023-09-15
?duration=30 // minutes
?serviceType=routine
```

### Response Payload:

json

```
{
  "date": "2023-09-15",
  "practitioner": "doc-uuid-456",
  "availableSlots": [
    {
      "start": "2023-09-15T09:00:00Z",
      "end": "2023-09-15T09:30:00Z",
      "type": "available"
    },
    {
      "start": "2023-09-15T10:00:00Z",
      "end": "2023-09-15T10:30:00Z",
      "type": "tentative"
    }
  ]
}
```

## 15. Waitlist Management

**Endpoint:** `POST /{appointmentId}/waitlist`

**Request Payload:**

```
json

{
  "patientId": "patient-uuid-123",
  "preferredDates": [
    "2023-09-15",
    "2023-09-16"
  ],
  "preferredTimes": ["morning", "afternoon"],
  "priority": "routine",
  "notificationPreferences": {
    "email": true,
    "sms": true,
    "advanceNotice": "24h"
  }
}
```

**Kafka Events:**

- Topic: `appointment.booked`
- Topic: `appointment.cancelled`
- Topic: `waitlist.added`



# User Auth Service

Base URL: `/api/v1/auth`

## 16. User Login

Endpoint: `POST /login`

### Request Payload:

```
json
{
  "username": "john.doe@example.com",
  "password": "SecurePass123!",
  "mfaCode": "123456",
  "deviceInfo": {
    "deviceId": "device-uuid-123",
    "userAgent": "Mozilla/5.0...",
    "ipAddress": "192.168.1.100"
  }
}
```

### Response Payload:

```
json
{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "refresh-token-uuid",
  "tokenType": "Bearer",
  "expiresIn": 3600,
  "user": {
    "id": "user-uuid-123",
    "email": "john.doe@example.com",
    "roles": ["patient"],
    "permissions": ["read:own_records", "write:own_profile"]
  }
}
```

### Validation Rules:

- Email format validation
- Password complexity requirements
- Rate limiting: 5 attempts per 15 minutes

- MFA required for privileged accounts
- Device registration for new devices

## 17. User Registration

**Endpoint:** POST /register

**Request Payload:**

```
json

{
  "email": "john.doe@example.com",
  "password": "SecurePass123!",
  "confirmPassword": "SecurePass123!",
  "profile": {
    "firstName": "John",
    "lastName": "Doe",
    "phone": "555-123-4567",
    "dateOfBirth": "1980-01-15"
  },
  "acceptTerms": true,
  "verificationMethod": "email"
}
```

## 18. Password Reset

**Endpoint:** POST /password-reset

**Request Payload:**

```
json

{
  "email": "john.doe@example.com",
  "resetMethod": "email"
}
```

**Response Payload:**

```
json
```

```
{
  "status": "sent",
  "message": "Password reset instructions sent to email",
  "resetTokenId": "reset-uuid-123",
  "expiresIn": 3600
}
```

## 19. Multi-Factor Authentication Setup

**Endpoint:** `POST /mfa/setup`

### Request Payload:

```
json

{
  "method": "totp",
  "deviceName": "iPhone 12"
}
```

### Response Payload:

```
json

{
  "qrCode": "...",
  "secret": "JBSWY3DPEHPK3PXP",
  "backupCodes": [
    "12345678",
    "87654321"
  ]
}
```

### Kafka Events:

- Topic: `user.authenticated`
  - Topic: `user.registered`
  - Topic: `security.failed_login`
-

# Role Management Service

Base URL: `/api/v1/roles`

## 20. Role Assignment

Endpoint: `POST /assign`

Request Payload:

```
json

{
  "userId": "user-uuid-123",
  "roles": ["doctor", "department_head"],
  "effectiveDate": "2023-09-01T00:00:00Z",
  "expiryDate": "2024-09-01T00:00:00Z",
  "assignedBy": "admin-uuid-456",
  "reason": "Promotion to department head"
}
```

Response Payload:

```
json

{
  "assignmentId": "assignment-uuid-789",
  "status": "active",
  "permissions": [
    "read:patient_records",
    "write:observations",
    "manage:department_users"
  ],
  "effectiveDate": "2023-09-01T00:00:00Z"
}
```

## 21. Permission Validation

Endpoint: `POST /validate`

Request Payload:

```
json
```

```
{
  "userId": "user-uuid-123",
  "resource": "Patient/patient-uuid-456",
  "action": "read",
  "context": {
    "facility": "facility-uuid-789",
    "department": "cardiology"
  }
}
```

### Response Payload:

```
json

{
  "allowed": true,
  "decision": "permit",
  "reason": "User has doctor role with read permission",
  "conditions": [
    "audit_required",
    "time_limited"
  ]
}
```

## 22. Attribute-Based Access Control

**Endpoint:** `POST /abac/evaluate`

### Request Payload:

```
json
```

```
{
  "subject": {
    "userId": "user-uuid-123",
    "roles": ["doctor"],
    "department": "cardiology",
    "facility": "facility-uuid-789"
  },
  "resource": {
    "type": "Patient",
    "id": "patient-uuid-456",
    "attributes": {
      "facility": "facility-uuid-789",
      "department": "cardiology"
    }
  },
  "action": "read",
  "environment": {
    "time": "2023-09-01T14:30:00Z",
    "location": "clinic"
  }
}
```

### Kafka Events:

- Topic: `role.assigned`
- Topic: `permission.denied`
- Topic: `access.granted`

## Telemedicine Service

Base URL: `/api/v1/telemedicine`

### 23. Session Creation

Endpoint: `POST /sessions`

#### Request Payload:

json

```
{
  "appointmentId": "appt-uuid-789",
  "participants": [
    {
      "userId": "patient-uuid-123",
      "role": "patient"
    },
    {
      "userId": "doc-uuid-456",
      "role": "provider"
    }
  ],
  "sessionType": "video_consultation",
  "scheduledStart": "2023-09-15T09:00:00Z",
  "estimatedDuration": 30
}
```

### Response Payload:

```
json

{
  "sessionId": "session-uuid-101",
  "joinUrls": {
    "patient": "https://telemedicine.example.com/join/patient-token-123",
    "provider": "https://telemedicine.example.com/join/provider-token-456"
  },
  "accessWindow": {
    "start": "2023-09-15T08:50:00Z",
    "end": "2023-09-15T09:40:00Z"
  },
  "sessionSettings": {
    "recordingEnabled": false,
    "chatEnabled": true,
    "screenShareEnabled": true
  }
}
```

## 24. Consent Management

**Endpoint:** `POST /consent`

### Request Payload:

```
json
```

```
{
  "sessionId": "session-uuid-101",
  "userId": "patient-uuid-123",
  "consentType": "video_recording",
  "granted": true,
  "timestamp": "2023-09-15T08:55:00Z",
  "ipAddress": "192.168.1.100"
}
```

## 25. Quality Monitoring

**Endpoint:** `GET /sessions/{sessionId}/metrics`

**Response Payload:**

```
json

{
  "sessionId": "session-uuid-101",
  "qualityMetrics": {
    "averageLatency": 45,
    "packetLoss": 0.2,
    "videoQuality": "HD",
    "audioQuality": "excellent"
  },
  "duration": 1800,
  "participants": [
    {
      "userId": "patient-uuid-123",
      "connectionTime": 1795,
      "disconnections": 0
    }
  ]
}
```

**Kafka Events:**

- Topic: `session.started`
  - Topic: `session.ended`
  - Topic: `consent.recorded`
-



# Notification Service

Base URL: `/api/v1/notifications`

## 26. Send Notification

Endpoint: `POST /send`

Request Payload:

```
json

{
  "recipientId": "user-uuid-123",
  "channels": ["email", "sms"],
  "template": "appointment_reminder",
  "data": {
    "appointmentDate": "2023-09-15T09:00:00Z",
    "doctorName": "Dr. Jane Smith",
    "location": "Room 101, Main Building"
  },
  "scheduledAt": "2023-09-14T20:00:00Z",
  "priority": "normal"
}
```

Response Payload:

```
json

{
  "notificationId": "notif-uuid-123",
  "status": "scheduled",
  "channels": [
    {
      "type": "email",
      "status": "queued",
      "estimatedDelivery": "2023-09-14T20:00:30Z"
    },
    {
      "type": "sms",
      "status": "queued",
      "estimatedDelivery": "2023-09-14T20:00:15Z"
    }
  ]
}
```

## 27. Template Management

**Endpoint:** POST /templates

**Request Payload:**

```
json

{
  "name": "appointment_reminder",
  "channels": {
    "email": {
      "subject": "Appointment Reminder - {{appointmentDate}}",
      "body": "Dear {{patientName}}, you have an appointment with {{doctorName}} on {{appointmentDate}}."
    },
    "sms": {
      "body": "Reminder: Appointment with {{doctorName}} on {{appointmentDate}}. Reply STOP to opt out."
    }
  },
  "variables": ["patientName", "doctorName", "appointmentDate"]
}
```

## 28. Bulk Notification

**Endpoint:** POST /bulk

**Request Payload:**

```
json

{
  "campaignName": "flu_shot_reminder",
  "template": "vaccination_reminder",
  "recipients": [
    {
      "userId": "user-uuid-123",
      "data": {"patientName": "John Doe"}
    },
    {
      "userId": "user-uuid-456",
      "data": {"patientName": "Jane Smith"}
    }
  ],
  "channels": ["email"],
  "scheduledAt": "2023-09-20T10:00:00Z"
}
```

**Kafka Events:**

- Topic: `notification.sent`
  - Topic: `notification.failed`
  - Topic: `notification.delivered`
- 

## Analytics Service

**Base URL:** `/api/v1/analytics`

### 29. Risk Scoring

**Endpoint:** `POST /risk-score`

#### Request Payload:

```
json

{
  "patientId": "patient-uuid-123",
  "riskType": "diabetes",
  "factors": [
    {
      "type": "observation",
      "code": "33747-0",
      "value": 95,
      "unit": "mg/dL"
    },
    {
      "type": "demographic",
      "age": 43,
      "gender": "male",
      "bmi": 28.5
    }
  ]
}
```

#### Response Payload:

```
json
```

```
{
  "patientId": "patient-uuid-123",
  "riskType": "diabetes",
  "score": 0.75,
  "level": "high",
  "confidence": 0.89,
  "factors": [
    {
      "name": "BMI",
      "contribution": 0.35,
      "weight": "high"
    },
    {
      "name": "Family History",
      "contribution": 0.25,
      "weight": "medium"
    }
  ],
  "recommendations": [
    "Regular glucose monitoring",
    "Dietary consultation"
  ],
  "calculatedAt": "2023-09-01T12:30:45Z"
}
```

## 41. ML Model Training Pipeline

**Endpoint:** `POST /ml/models/train`

**Request Payload:**

json

```
{
  "trainingJob": {
    "modelName": "diabetes_risk_classifier",
    "modelType": "classification",
    "algorithm": "random_forest",
    "targetVariable": "diabetes_diagnosis",
    "features": [
      {
        "name": "age",
        "type": "numeric",
        "source": "Patient.birthDate",
        "preprocessing": "age_calculation"
      },
      {
        "name": "bmi",
        "type": "numeric",
        "source": "Observation",
        "loincCode": "39156-5",
        "preprocessing": "outlier_removal"
      },
      {
        "name": "glucose_level",
        "type": "numeric",
        "source": "Observation",
        "loincCode": "2339-0",
        "aggregation": "latest_value"
      },
      {
        "name": "family_history",
        "type": "categorical",
        "source": "FamilyMemberHistory",
        "valueSet": "diabetes_conditions"
      }
    ],
    "trainingParameters": {
      "dataRange": {
        "start": "2020-01-01",
        "end": "2023-08-31"
      },
      "validationSplit": 0.2,
      "testSplit": 0.1,
      "hyperparameters": {
        "n_estimators": 100,
        "max_depth": 10,
        "min_samples_split": 5,
        "random_state": 42
      }
    }
  }
}
```

```
    },
    "crossValidation": {
      "folds": 5,
      "stratified": true
    }
  },
  "dataPrivacy": {
    "anonymization": "k_anonymity",
    "k_value": 5,
    "excludePII": true,
    "auditRequired": true
  }
}
```

## Response Payload:

```
json

{
  "trainingJobId": "job-uuid-123",
  "status": "running",
  "estimatedCompletion": "2023-09-01T14:30:45Z",
  "datasetInfo": {
    "totalRecords": 15420,
    "trainingRecords": 12336,
    "validationRecords": 1542,
    "testRecords": 1542,
    "featureCount": 15,
    "classDistribution": {
      "diabetes": 0.23,
      "pre_diabetes": 0.31,
      "normal": 0.46
    }
  },
  "progress": {
    "currentStep": "feature_engineering",
    "completionPercent": 25,
    "estimatedTimeRemaining": "PT45M"
  }
}
```

## Validation Rules:

- Minimum dataset size: 1000 records
- Feature correlation analysis required

- Data quality validation mandatory
- Privacy compliance verification
- Model interpretability assessment

## 42. Personalized Clinical Alerts

**Endpoint:** `POST /ml/alerts/personalized`

### Request Payload:

```
json

{
  "patientId": "patient-uuid-123",
  "modelId": "model-uuid-456",
  "alertConfiguration": {
    "riskThreshold": 0.75,
    "alertTypes": ["high_risk", "trend_deterioration"],
    "timeHorizon": "P30D",
    "updateFrequency": "daily"
  },
  "contextualFactors": [
    {
      "type": "recent_observations",
      "lookbackPeriod": "P7D",
      "weight": 0.4
    },
    {
      "type": "medication_adherence",
      "source": "MedicationStatement",
      "weight": 0.3
    },
    {
      "type": "lifestyle_factors",
      "source": "Observation",
      "categories": ["exercise", "diet", "smoking"],
      "weight": 0.3
    }
  ]
}
```

### Response Payload:

```
json
```

```
{
  "alertId": "alert-uuid-789",
  "patientId": "patient-uuid-123",
  "riskAssessment": {
    "overallRisk": 0.82,
    "riskLevel": "high",
    "confidence": 0.89,
    "prediction": {
      "condition": "type_2_diabetes",
      "probability": 0.82,
      "timeHorizon": "P30D"
    }
  },
  "contributingFactors": [
    {
      "factor": "elevated_glucose",
      "impact": 0.35,
      "recentTrend": "increasing",
      "lastValue": 145,
      "referenceRange": "70-100 mg/dL"
    },
    {
      "factor": "bmi",
      "impact": 0.28,
      "value": 32.5,
      "category": "obese"
    }
  ],
  "recommendations": [
    {
      "type": "clinical_action",
      "priority": "high",
      "action": "Schedule endocrinology consultation",
      "reasoning": "High diabetes risk with recent glucose elevation"
    },
    {
      "type": "lifestyle_intervention",
      "priority": "medium",
      "action": "Initiate dietary counseling",
      "evidenceLevel": "strong"
    }
  ],
  "fhirResources": [
    {
      "resourceType": "RiskAssessment",
      "id": "risk-assess-uuid-101",
```



```
    "status": "final"
  }
]
}
```

## 43. Model Versioning and Lifecycle

**Endpoint:** `POST /ml/models/{modelId}/versions`

**Request Payload:**

```
json

{
  "versionInfo": {
    "version": "2.1.0",
    "description": "Improved diabetes risk model with additional lifestyle factors",
    "trainingJobId": "job-uuid-123",
    "baselineModel": "model-uuid-456-v2.0.0"
  },
  "performanceMetrics": {
    "accuracy": 0.89,
    "precision": 0.87,
    "recall": 0.91,
    "f1Score": 0.89,
    "auc": 0.94,
    "crossValidationScore": 0.88
  },
  "modelArtifacts": {
    "modelFile": "diabetes_rf_v2.1.0.pkl",
    "featureImportance": "feature_importance_v2.1.0.json",
    "preprocessor": "preprocessor_v2.1.0.pkl",
    "metadata": "model_metadata_v2.1.0.json"
  },
  "deploymentConfig": {
    "environment": "production",
    "rolloutStrategy": "blue_green",
    "canaryPercentage": 10,
    "monitoringPeriod": "P7D"
  }
}
```

**Response Payload:**

```
json
```

```
{
  "modelVersionId": "model-version-uuid-456",
  "status": "deployed",
  "deploymentTimestamp": "2023-09-01T12:30:45Z",
  "performanceComparison": {
    "previousVersion": {
      "version": "2.0.0",
      "accuracy": 0.85,
      "auc": 0.91
    },
    "improvement": {
      "accuracy": 0.04,
      "auc": 0.03,
      "statisticalSignificance": true
    }
  },
  "productionMetrics": {
    "predictionLatency": "PT0.05S",
    "throughput": "1000/minute",
    "errorRate": 0.001
  }
}
```

## 44. Healthcare Trend Analysis

**Endpoint:** `GET /analytics/trends`

### Query Parameters:

```
?metric=readmission_rate,infection_rate,mortality_rate
?timeRange=P1Y
?granularity=monthly
?demographics=age,gender,department
?facility=facility-uuid-001
```

### Response Payload:

json

```
{
  "analysisId": "trend-analysis-uuid-789",
  "timeRange": {
    "start": "2022-09-01T00:00:00Z",
    "end": "2023-09-01T00:00:00Z"
  },
  "trends": [
    {
      "metric": "readmission_rate",
      "overall": {
        "currentValue": 0.12,
        "previousPeriod": 0.15,
        "changePercent": -20.0,
        "trend": "decreasing",
        "significance": "p < 0.05"
      }
    },
    "byDemographics": [
      {
        "segment": "age_65_plus",
        "value": 0.18,
        "trend": "stable",
        "sampleSize": 1247
      },
      {
        "segment": "cardiology_dept",
        "value": 0.08,
        "trend": "decreasing",
        "sampleSize": 892
      }
    ],
    "timeSeriesData": [
      {
        "period": "2022-09",
        "value": 0.15,
        "confidenceInterval": [0.13, 0.17]
      },
      {
        "period": "2023-08",
        "value": 0.12,
        "confidenceInterval": [0.10, 0.14]
      }
    ]
  },
  "correlations": [
    {
```

```
"metric1": "readmission_rate",
"metric2": "average_length_of_stay",
"correlation": -0.65,
"significance": "p < 0.001"
}
],
"anomalies": [
{
"metric": "infection_rate",
"period": "2023-06",
"value": 0.08,
"expected": 0.05,
"zScore": 2.3,
"investigation": "required"
}
]
}
```

## 45. FHIR Resource Linking for Predictions

**Endpoint:** `POST /ml/predictions/link-fhir`

**Request Payload:**

json

```
{
  "predictionId": "prediction-uuid-123",
  "patientId": "patient-uuid-456",
  "modelId": "model-uuid-789",
  "fhirMapping": {
    "primaryResource": {
      "resourceType": "RiskAssessment",
      "method": {
        "coding": [
          {
            "system": "http://example.org/ml-models",
            "code": "diabetes_risk_rf_v2.1.0",
            "display": "Diabetes Risk Random Forest v2.1.0"
          }
        ]
      },
    },
  },
  "prediction": [
    {
      "outcome": {
        "coding": [
          {
            "system": "http://snomed.info/sct",
            "code": "44054006",
            "display": "Type 2 diabetes mellitus"
          }
        ]
      },
    },
  ],
  "probabilityDecimal": 0.82,
  "whenRange": {
    "low": {
      "value": 30,
      "unit": "d"
    },
    "high": {
      "value": 90,
      "unit": "d"
    }
  },
},
"supportingResources": [
  {
    "resourceType": "DetectedIssue",
    "status": "final",
    "category": {
```

```
"coding": [  
  {  
    "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",  
    "code": "CAUTION"  
  }  
],  
"detail": "High risk for developing diabetes based on current health indicators"  
}  
]  
}  
}
```

## Response Payload:

json

```

{
  "linkingId": "fhir-link-uuid-101",
  "fhirResources": [
    {
      "resourceType": "RiskAssessment",
      "id": "risk-assess-uuid-202",
      "status": "final",
      "subject": {
        "reference": "Patient/patient-uuid-456"
      },
      "performer": {
        "reference": "Device/ml-model-device-uuid-303"
      },
      "prediction": [
        {
          "outcome": {
            "coding": [
              {
                "system": "http://snomed.info/sct",
                "code": "44054006",
                "display": "Type 2 diabetes mellitus"
              }
            ]
          },
          "probabilityDecimal": 0.82
        }
      ]
    }
  ],
  "auditTrail": {
    "createdBy": "ml-system",
    "createdAt": "2023-09-01T12:30:45Z",
    "modelVersion": "2.1.0",
    "inputFeatures": 15,
    "confidence": 0.89
  }
}

```

### Validation Rules:

- Model training data minimum quality thresholds
- Feature engineering validation requirements
- Model performance benchmarking mandatory
- Bias detection and fairness assessment

- Clinical validation by domain experts
- Real-time prediction latency requirements
- FHIR resource consistency validation
- Audit trail completeness verification

### Kafka Events:

- Topic: `model.trained`
  - Topic: `risk.calculated`
  - Topic: `alert.generated`
- 

## Audit Logging Service

**Base URL:** `/api/v1/audit`

### 32. Log Access Event

**Endpoint:** `POST /events`

#### Request Payload:

```
json
{
  "eventType": "data_access",
  "userId": "user-uuid-123",
  "resourceType": "Patient",
  "resourceId": "patient-uuid-456",
  "action": "read",
  "timestamp": "2023-09-01T12:30:45Z",
  "sessionId": "session-uuid-789",
  "ipAddress": "192.168.1.100",
  "userAgent": "Mozilla/5.0...",
  "metadata": {
    "reason": "Patient consultation",
    "department": "cardiology",
    "facility": "facility-uuid-001"
  }
}
```

#### Response Payload:

```
json
```



```
{
  "auditId": "audit-uuid-123",
  "status": "logged",
  "timestamp": "2023-09-01T12:30:45Z",
  "immutableHash": "sha256:abc123def456..."
}
```

### 33. Export Audit Logs

**Endpoint:** GET /export

**Query Parameters:**

```
?startDate=2023-08-01
?endDate=2023-09-01
?userId=user-uuid-123
?resourceType=Patient
?format=csv
```

**Response Payload:**

```
json

{
  "exportId": "export-uuid-456",
  "status": "processing",
  "format": "csv",
  "downloadUrl": "/api/v1/audit/exports/export-uuid-456",
  "estimatedCompletion": "2023-09-01T12:35:45Z",
  "digitalSignature": "MIIC..."
}
```

### 34. Anomaly Detection

**Endpoint:** GET /anomalies

**Response Payload:**

```
json
```

```
{
  "period": "P7D",
  "anomalies": [
    {
      "type": "unusual_access_pattern",
      "userId": "user-uuid-123",
      "description": "Access to 50+ patient records in 1 hour",
      "severity": "medium",
      "timestamp": "2023-09-01T02:00:00Z",
      "score": 0.75
    }
  ]
}
```

### Kafka Events:

- Topic: `audit.logged`
- Topic: `audit.anomaly_detected`
- Topic: `audit.export_requested`

## FHIR API Gateway

Base URL: `/fhir/R4`

### 35. External FHIR Access

Endpoint: `GET /Patient/{patientId}`

#### Request Headers:

Authorization: Bearer {api\_key}  
Accept: application/fhir+json

#### Response Payload:

json

```
{
  "resourceType": "Patient",
  "id": "patient-uuid-123",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2023-09-01T12:30:45Z"
  },
  "identifier": [
    {
      "use": "usual",
      "value": "MRN12345"
    }
  ]
}
```

## 36. Capability Statement

**Endpoint:** GET /metadata

**Response Payload:**

json

```
{
  "resourceType": "CapabilityStatement",
  "status": "active",
  "date": "2023-09-01",
  "publisher": "Healthcare Organization",
  "kind": "instance",
  "software": {
    "name": "FHIR Patient Portal",
    "version": "1.0.0"
  },
  "fhirVersion": "4.0.1",
  "format": ["json", "xml"],
  "rest": [
    {
      "mode": "server",
      "resource": [
        {
          "type": "Patient",
          "interaction": [
            {"code": "read"},
            {"code": "search-type"}
          ]
        }
      ]
    }
  ]
}
```

## 37. Batch Operations

Endpoint: POST /

Request Payload:

json

```
{
  "resourceType": "Bundle",
  "type": "batch",
  "entry": [
    {
      "request": {
        "method": "GET",
        "url": "Patient/patient-uuid-123"
      }
    },
    {
      "request": {
        "method": "GET",
        "url": "Observation?patient=patient-uuid-123"
      }
    }
  ]
}
```

### Kafka Events:

- Topic: `fhir.request`
- Topic: `fhir.response`
- Topic: `gateway.rate_limited`

---

## Kafka Communication Patterns (Enhanced)

### Topic Architecture and Naming Conventions

#### Topic Structure:

```
{service}.{entity}.{action}.{version}
```

#### Examples:

patient.registered.v1  
observation.created.v1  
appointment.cancelled.v1  
audit.logged.v1  
hl7.message.received.v1  
fhir.resource.created.v1  
security.alert.triggered.v1  
ml.prediction.generated.v1

Core Event Schema

Base Event Structure:

json

```
{
  "eventId": "event-uuid-123",
  "eventType": "patient.registered.v1",
  "eventVersion": "1.0",
  "timestamp": "2023-09-01T12:30:45Z",
  "source": {
    "service": "patient-service",
    "version": "2.1.0",
    "instance": "patient-service-pod-3"
  },
  "subject": {
    "type": "Patient",
    "id": "patient-uuid-123",
    "tenantId": "tenant-clinic-001"
  },
  "data": {
    "action": "created",
    "userId": "user-uuid-456",
    "previousState": null,
    "currentState": {
      "status": "active",
      "registrationComplete": true
    }
  },
  "metadata": {
    "correlationId": "corr-uuid-789",
    "causationId": "cause-uuid-101",
    "sessionId": "session-uuid-202",
    "traceId": "trace-uuid-303",
    "priority": "normal",
    "retryCount": 0
  },
  "compliance": {
    "dataClassification": "PHI",
    "retentionPeriod": "P7Y",
    "encryptionRequired": true,
    "auditRequired": true
  }
}
```

## Topic Configurations

### 1. Patient Service Topics

Topic: **patient.lifecycle.v1**

json

```
{
  "partitions": 12,
  "replicationFactor": 3,
  "retentionMs": 604800000,
  "partitionKey": "patientId",
  "compressionType": "lz4",
  "cleanupPolicy": "delete"
}
```

## Message Types:

- patient.registered.v1
- patient.updated.v1
- patient.merged.v1
- patient.anonymized.v1
- patient.consent.changed.v1

Topic: patient.search.v1

json

```
{
  "partitions": 6,
  "replicationFactor": 3,
  "retentionMs": 86400000,
  "partitionKey": "searchHash",
  "compressionType": "snappy"
}
```

## 2. Clinical Data Topics

Topic: observation.clinical.v1

json

```
{
  "partitions": 24,
  "replicationFactor": 3,
  "retentionMs": 2592000000,
  "partitionKey": "patientId",
  "compressionType": "lz4"
}
```



## Message Types:

- `observation.created.v1`
- `observation.updated.v1`
- `observation.alert.triggered.v1`
- `observation.batch.processed.v1`

**Topic:** `appointment.scheduling.v1`

```
json

{
  "partitions": 8,
  "replicationFactor": 3,
  "retentionMs": 1209600000,
  "partitionKey": "practitionerId",
  "compressionType": "lz4"
}
```

## 3. Security and Audit Topics

**Topic:** `security.events.v1`

```
json

{
  "partitions": 16,
  "replicationFactor": 3,
  "retentionMs": 31536000000,
  "partitionKey": "userId",
  "compressionType": "gzip",
  "cleanupPolicy": "compact"
}
```

**Topic:** `audit.trail.v1`

```
json
```

```
{
  "partitions": 32,
  "replicationFactor": 3,
  "retentionMs": 220752000000,
  "partitionKey": "resourceId",
  "compressionType": "gzip",
  "cleanupPolicy": "compact"
}
```

## Dead Letter Queue Configuration

Topic: `dlq.failed-events.v1`

json

```
{
  "partitions": 4,
  "replicationFactor": 3,
  "retentionMs": 604800000,
  "compressionType": "gzip",
  "retryPolicy": {
    "maxRetries": 3,
    "backoffStrategy": "exponential",
    "initialDelay": "PT1S",
    "maxDelay": "PT30S",
    "jitterEnabled": true
  },
  "alerting": {
    "enabled": true,
    "threshold": 100,
    "timeWindow": "PT1H",
    "channels": ["slack", "email"]
  }
}
```

## DLQ Message Schema:

json

```
{
  "dlqEventId": "dlq-event-uuid-456",
  "originalEvent": {
    "eventId": "original-event-uuid-123",
    "eventType": "patient.registered.v1",
    "originalPayload": "base64EncodedOriginalMessage"
  },
  "failureInfo": {
    "failureReason": "InvalidPatientData",
    "errorMessage": "Required field 'birthDate' is missing",
    "failureTimestamp": "2023-09-01T12:30:45Z",
    "retryAttempts": 3,
    "lastRetryTimestamp": "2023-09-01T12:33:45Z"
  },
  "routing": {
    "originalTopic": "patient.lifecycle.v1",
    "originalPartition": 5,
    "failedConsumer": "patient-service-consumer-2"
  }
}
```

## Event Ordering and Partitioning Strategy

### Partitioning Rules:

1. **Patient Events:** Partition by `patientId` to ensure ordering
2. **User Events:** Partition by `userId` for session consistency
3. **Appointment Events:** Partition by `practitionerId` for schedule consistency
4. **Audit Events:** Partition by `resourceId` for resource-level ordering
5. **Notification Events:** Partition by `recipientId` for delivery order

### Event Ordering Example:

json

```

{
  "partitioningStrategy": {
    "patient.lifecycle.v1": {
      "partitionKey": "data.patientId",
      "keyExtractor": "$.subject.id",
      "orderingGuarantee": "per_patient"
    },
    "observation.clinical.v1": {
      "partitionKey": "data.patientId",
      "keyExtractor": "$.data.subject.reference",
      "orderingGuarantee": "per_patient_per_observation_type"
    }
  }
}

```

## Consumer Groups and Processing Patterns

### 1. Real-time Processing Consumers

#### Patient Service Consumer:

```

json

{
  "groupId": "patient-service-realtime",
  "topics": [
    "hl7.message.received.v1",
    "user.registered.v1",
    "consent.updated.v1"
  ],
  "processingMode": "exactly_once",
  "maxPollRecords": 100,
  "sessionTimeoutMs": 30000,
  "autoCommit": false
}

```

#### Notification Service Consumer:

```

json

```

```
{
  "groupId": "notification-service-immediate",
  "topics": [
    "appointment.booked.v1",
    "observation.alert.triggered.v1",
    "security.alert.triggered.v1"
  ],
  "processingMode": "at_least_once",
  "maxPollRecords": 50,
  "priorityQueues": {
    "critical": ["security.alert.triggered.v1"],
    "high": ["observation.alert.triggered.v1"],
    "normal": ["appointment.booked.v1"]
  }
}
```

## 2. Batch Processing Consumers

### Analytics Service Consumer:

```
json

{
  "groupId": "analytics-service-batch",
  "topics": [
    "patient.lifecycle.v1",
    "observation.clinical.v1",
    "appointment.scheduling.v1"
  ],
  "processingMode": "batch",
  "batchSize": 1000,
  "batchTimeoutMs": 60000,
  "windowDuration": "PT5M"
}
```

## 3. Audit and Compliance Consumers

### Audit Service Consumer:

```
json
```

```
{
  "groupId": "audit-service-compliance",
  "topics": [
    "/*.v1"
  ],
  "processingMode": "exactly_once",
  "durabilityGuarantee": "persistent",
  "retentionPolicy": "P7Y",
  "encryptionEnabled": true
}
```

## Error Handling and Retry Patterns

### Retry Configuration:

```
json

{
  "retryPolicy": {
    "retryableExceptions": [
      "org.apache.kafka.common.errors.TimeoutException",
      "org.springframework.dao.TransientDataAccessException",
      "java.net.SocketTimeoutException"
    ],
    "nonRetryableExceptions": [
      "com.fhir.validation.ValidationException",
      "org.springframework.security.access.AccessDeniedException",
      "com.fhir.patient.PatientNotFoundException"
    ],
    "maxRetries": 3,
    "backoffPolicy": {
      "type": "exponential",
      "initialInterval": 1000,
      "multiplier": 2.0,
      "maxInterval": 30000,
      "randomizationFactor": 0.1
    }
  }
}
```

## Performance Monitoring and Alerting

### Kafka Metrics Collection:

```
json
```

```
{
  "monitoring": {
    "metricsReporter": "io.confluent.monitoring.clients.interceptor.MonitoringProducerInterceptor",
    "jmxMetrics": [
      "kafka.producer:type=producer-metrics,client-id=*",
      "kafka.consumer:type=consumer-metrics,client-id=*",
      "kafka.streams:type=stream-metrics,client-id="
    ],
    "alertRules": [
      {
        "metric": "consumer_lag",
        "threshold": 10000,
        "duration": "PT5M",
        "severity": "warning"
      },
      {
        "metric": "error_rate",
        "threshold": 0.05,
        "duration": "PT2M",
        "severity": "critical"
      }
    ]
  }
}
```

## Topic Health Monitoring:

json

```

{
  "healthChecks": {
    "producerLatency": {
      "threshold": "PT0.1S",
      "alertChannel": "ops-team"
    },
    "consumerLag": {
      "threshold": 5000,
      "alertChannel": "dev-team"
    },
    "diskUsage": {
      "threshold": 0.8,
      "alertChannel": "infrastructure"
    },
    "replicationStatus": {
      "minInSyncReplicas": 2,
      "alertChannel": "ops-team"
    }
  }
}

```

## Schema Evolution and Versioning

### Schema Registry Configuration:

```

json

{
  "schemaRegistry": {
    "url": "http://schema-registry:8081",
    "compatibilityLevel": "BACKWARD",
    "subjectNaming": "TopicNameStrategy",
    "schemas": {
      "patient.lifecycle.v1-value": {
        "version": 2,
        "evolution": "backward_compatible",
        "changes": [
          "added optional field 'preferredLanguage'",
          "added optional field 'communicationPreferences'"
        ]
      }
    }
  }
}

```

## Security Configuration



## Kafka Security Settings:

```
json
{
  "security": {
    "protocol": "SASL_SSL",
    "saslMechanism": "SCRAM-SHA-512",
    "sslTruststoreLocation": "/opt/kafka/ssl/truststore.jks",
    "sslKeystoreLocation": "/opt/kafka/ssl/keystore.jks",
    "encryption": {
      "inTransit": "TLS 1.3",
      "atRest": "AES-256-GCM"
    },
  },
  "acls": [
    {
      "principal": "User:patient-service",
      "operations": ["Read", "Write"],
      "topics": ["patient.lifecycle.v1"]
    },
    {
      "principal": "User:audit-service",
      "operations": ["Read"],
      "topics": ["**.*.v1"]
    }
  ]
}
```

This comprehensive Kafka configuration ensures:

- **High Availability:** Multi-partition, multi-replica topics
- **Data Consistency:** Exactly-once processing where required
- **Scalability:** Partitioning strategies for horizontal scaling
- **Security:** End-to-end encryption and access controls
- **Compliance:** Audit trails and retention policies
- **Monitoring:** Real-time metrics and alerting
- **Error Handling:** Robust retry mechanisms and dead letter queues
- **Schema Evolution:** Backward-compatible schema changes

The event-driven architecture enables loose coupling between microservices while maintaining data consistency and providing comprehensive audit trails for healthcare compliance requirements.