Miami Boundary Scenario Implementation (v0.52)

Overview

Implemented a new “Miami Boundary Scenario” view that uses precise KML boundary lines to assign territories and displays individual customer accounts as color-coded dots on the map.

KML Boundary Parsing

Source File

- **File**: `Miami Territory Breakout.kml` (uploaded by user)
- **Boundaries**: NORTH BOUNDARY and SOUTH BOUNDARY polylines
- **Format**: Google Earth KML with LineString coordinates

Boundary Lines Extracted

NORTH BOUNDARY

- **Points**: 34 coordinate pairs
- **Longitude Range**: -80.121 to -80.385
- **Latitude Range**: ~25.896 to 25.901
- **Color**: Blue (#3B82F6)
- **Description**: Accounts north of this line → Miami - North

SOUTH BOUNDARY

- **Points**: 18 coordinate pairs
- **Longitude Range**: -80.119 to -80.387
- **Latitude Range**: ~25.840 to 25.856
- **Color**: Orange (#F59E0B)
- **Description**: Accounts south of this line → Miami - South

---

Territory Assignment Algorithm

Logic

```
function assignTerritory(point, boundaries) {
  const isNorth = isNorthOfBoundary(point, boundaries.north);
  const isSouth = !isNorthOfBoundary(point, boundaries.south);

  if (isNorth) return 'North';
  else if (isSouth) return 'South';
  else return 'Central';
}
```

Boundary Interpolation

- For each customer location, find the boundary line segment that brackets the longitude
- Interpolate the boundary latitude at that longitude
- Compare customer latitude with interpolated boundary latitude
- Account is “north” if its latitude > boundary latitude

KML Scenario Results

Territory Distribution

Territory	Accounts	vs. Original	Change
North	202	282 (original)	-80 accounts
Central	357	312 (original)	+45 accounts
South	315	280 (original)	+35 accounts
Total	874	874	0

Comparison with Original Assignment

Original Method: Simple latitude threshold (25.89 for north, 25.843 for south)

KML Method: Precise polyline boundaries from Google Earth

Account Movements

- **North → Central**: 80 accounts
- **Central → South**: 35 accounts
- **Unchanged**: 759 accounts (87%)
- **Total Changes**: 115 accounts (13%)

Why the Differences?

1. **Original boundaries** were straight horizontal lines (constant latitude)
2. **KML boundaries** follow actual roads/highways with curves and turns
3. KML boundaries are more precise and follow geographic features
4. Some accounts near boundary lines got reassigned due to road alignment

---

Data Files Generated

1. miami-kml-scenario.json

- **Location**: `/nextjs_space/public/miami-kml-scenario.json`
- **Records**: 874 customer accounts
- **New Fields**:
- `kmlTerritory`: Territory based on KML boundaries

- `originalTerritory`: Original territory assignment
- **Purpose**: Display individual accounts in scenario view

2. miami-kml-boundaries.json

- **Location**: `/nextjs_space/public/miami-kml-boundaries.json`
- **Content**:
  ```
  json
  {
    "north": [{lng, lat}, ...],  // 34 points
    "south": [{lng, lat}, ...]   // 18 points
  }
  ```
- **Purpose**: Render boundary polylines on map

Component Implementation

New Component: MiamiKMLScenarioView

- **File**: `/components/miami-kml-scenario-view.tsx`
- **Purpose**: Display KML boundary scenario with individual account dots

Features

1. **Individual Account Markers**
   - Each customer displayed as a circle dot (not ZIP polygon)
   - Color-coded by KML territory assignment
   - Scale: 6px normal, 10px when highlighted
   - Opacity: 0.8 normal, 1.0 when highlighted

2. **Boundary Polylines**
   - North boundary: Blue line, 4px width
   - South boundary: Orange line, 4px width
   - Geodesic rendering for accurate curvature

3. **Interactive Features**
   - Click any dot to see customer details
   - Search by account number, name, or address
   - Office location marker (gold circle)
   - InfoWindow with full customer data

4. **Territory Filters**
   - Toggle North/Central/South visibility
   - Filter buttons inherited from parent component
   - Real-time dot filtering

5. **Customer InfoWindow Details**
   - Account number
   - Customer name
   - Full address
   - KML territory assignment
   - Original territory (if changed)

- Monthly price
- Service route

UI Integration

New View Mode Button

- **Label**: “Miami Boundary Scenario”
- **Icon**: MapPin
- **Color**: Indigo (#4F46E5)
- **Visibility**: Only shown when location = ‘miami’
- **Position**: Between “Residential Account Territory Assignments” and “Density Analysis”

Navigation Path

1. Select “Miami, FL” from location dropdown
2. Click “Miami Boundary Scenario” button
3. View individual accounts as colored dots
4. See KML boundary lines overlaid on map
5. Use filters to toggle territories

Technical Details

Processing Script

- **File**: `/parse_miami_kml_boundaries.js`
- **Dependencies**: xml2js for KML parsing
- **Process**:
  1. Parse KML file using xml2js
  2. Extract NORTH BOUNDARY and SOUTH BOUNDARY coordinates
  3. Load miami-route-assignments.json
  4. For each customer, determine territory using interpolation
  5. Generate miami-kml-scenario.json with kmlTerritory field
  6. Save boundary lines to miami-kml-boundaries.json
  7. Output statistics and change summary

Interpolation Algorithm

```
function isNorthOfBoundary(point, boundaryLine) {
  // Find boundary segment at point's longitude
  let leftPoint, rightPoint;
  for (let i = 0; i < boundaryLine.length - 1; i++) {
    if (within_longitude_range(point, boundaryLine[i], boundaryLine[i+1])) {
      leftPoint = boundaryLine[i];
      rightPoint = boundaryLine[i+1];
      break;
    }
  }

  // Interpolate boundary latitude
  const fraction = (point.lng - leftPoint.lng) / (rightPoint.lng - leftPoint.lng);
  const boundaryLat = leftPoint.lat + fraction * (rightPoint.lat - leftPoint.lat);

  // Compare
  return point.lat > boundaryLat;
}
```

Visual Design

Color Scheme

- **North Territory**: Blue (#3B82F6)
- **Central Territory**: Green (#10B981)
- **South Territory**: Orange (#F59E0B)
- **Office Marker**: Gold (#FFD700)
- **Boundary Lines**: Same as territory colors

Marker Styling

- **Shape**: Circle (google.maps.SymbolPath.CIRCLE)
- **Size**: 6px (normal), 10px (highlighted)
- **Fill Opacity**: 0.8 (normal), 1.0 (highlighted)
- **Stroke**: 1px white (normal), 3px black (highlighted)
- **Stroke Opacity**: 1.0

Boundary Line Styling

- **Width**: 4px
- **Opacity**: 0.8
- **Type**: Geodesic (curves with earth’s surface)
- **Rendering**: PolylineF component

Comparison: Original vs KML Scenario

Original Territory View (v0.51)

- Uses ZIP code polygons

- Territories based on latitude thresholds
- Shows aggregated ZIP data
- 282 North, 312 Central, 280 South

KML Boundary Scenario (v0.52)

- Uses individual account dots
- Territories based on KML polylines
- Shows individual customer locations
- 202 North, 357 Central, 315 South

Visual Differences

Feature	Original	KML Scenario
Display	ZIP polygons	Individual dots
Boundaries	Implicit (by ZIP)	Explicit polylines
Territory Assignment	Latitude threshold	KML line interpolation
Granularity	ZIP-level	Account-level
Search	ZIP code	Customer/account
Click Info	ZIP summary	Individual customer

Use Cases

Business Planning

1. **Precise Territory Delineation**: See exact boundary lines drawn in Google Earth
2. **Account Distribution**: View individual customer locations, not aggregated ZIPs
3. **Impact Analysis**: Compare KML scenario vs original to see account movements
4. **Field Validation**: Click individual accounts to verify territory assignments

Operational Planning

1. **Route Optimization**: See actual customer locations for route planning
2. **Territory Balancing**: Compare account counts between scenarios
3. **Boundary Refinement**: Identify accounts near boundary lines for review
4. **Service Area Definition**: Validate territory boundaries match operational needs

---

Statistics and Insights

Territory Balance Analysis

Original Distribution

- North: 282 accounts (32.3%)

- Central: 312 accounts (35.7%)
- South: 280 accounts (32.0%)
- **Balance**: Relatively even (±3.7% from average)

KML Distribution

- North: 202 accounts (23.1%)
- Central: 357 accounts (40.8%)
- South: 315 accounts (36.1%)
- **Balance**: Less even (Central is largest)

Recommendations

1. **Consider Central Split**: Central has 40.8% of accounts (357)
2. **North Territory Review**: Only 23.1% of accounts (202) - may need expansion
3. **Boundary Adjustment**: Review KML boundaries to better balance territories
4. **Hybrid Approach**: Consider adjusting KML lines to balance workload

---

Files Created/Modified

Created

- `/parse_miami_kml_boundaries.js` - KML parsing script
- `/components/miami-kml-scenario-view.tsx` - React component
- `/public/miami-kml-scenario.json` - Account data with KML territories
- `/public/miami-kml-boundaries.json` - Boundary polyline coordinates
- `MIAMI_BOUNDARY_AND_COLOR_ENHANCEMENTS.md` - This documentation

Modified

- `/components/territory-map.tsx` - Added KML scenario view mode
- `/app/page.tsx` - Updated version to v0.52
- `/package.json` - Added xml2js dependency (indirectly via yarn)

Testing Checklist

- [x] KML file parsed successfully (34 + 18 points)
- [x] All 874 accounts processed and assigned
- [x] Territory distribution calculated correctly
- [x] miami-kml-scenario.json generated (874 records)
- [x] miami-kml-boundaries.json generated (2 polylines)
- [x] MiamiKMLScenarioView component renders
- [x] Individual account dots display correctly
- [x] Boundary polylines render on map
- [x] Territory colors match specification
- [x] Search functionality works
- [x] InfoWindows display customer details
- [x] Office location marker displays

- [x] Territory filters functional
- [x] View mode button appears for Miami only
- [x] Build completes successfully
- [x] No TypeScript errors

Deployment Notes

Prerequisites

- xml2js package installed (yarn add xml2js)
- KML boundary file processed
- Data files in /public directory

Build Process

```
cd /home/ubuntu/phoenix_territory_map/nextjs_space
yarn build
```

Bundle Size

- **Before**: 197 KB (First Load JS)
- **After**: 198 KB (First Load JS)
- **Increase**: +1 KB (+0.5%)

Future Enhancements

Potential Features

1. **Boundary Editing**: Allow admins to adjust boundary lines
2. **What-If Analysis**: Create multiple boundary scenarios
3. **Revenue Overlay**: Color dots by account value instead of territory
4. **Route Visualization**: Show technician routes overlaid on dots
5. **Clustering**: Group nearby dots at lower zoom levels
6. **Export Options**: Download KML scenario data
7. **Comparison View**: Side-by-side original vs KML
8. **Territory Metrics**: Show drive time, workload balance

Data Enhancements

1. **Historical Tracking**: Compare scenarios over time
2. **Performance Metrics**: Territory efficiency analysis
3. **Customer Attributes**: Filter by service type, price range
4. **Geographic Analysis**: Distance from office, drive time zones

---

Contact Information

Central Office: APS of Miami
Address: 11720 Biscayne Blvd, Miami, FL 33181

KML Scenario Distribution:
- North: 202 accounts (23.1%)
- Central: 357 accounts (40.8%)
- South: 315 accounts (36.1%)

Questions: Contact sjohnson@amenitypool.com

Version History

v0.52 (December 31, 2025)
- ✅ KML boundary file parsing
- ✅ Precise territory assignment using polyline interpolation
- ✅ Individual account dot visualization
- ✅ Boundary polylines rendered on map
- ✅ New “Miami Boundary Scenario” view mode
- ✅ Customer search and InfoWindow details
- ✅ Territory comparison statistics
- ✅ 202 North, 357 Central, 315 South accounts

v0.51 (December 31, 2025)
- Location selector in territory view
- Miami summary statistics
- Enhanced territory filters

v0.50 (December 30, 2025)
- Initial Miami territory implementation
- ZIP-based territory assignments
- Interactive map with boundaries

Document generated: December 31, 2025
Application Version: v0.52
Status: ✅ Production Ready