ROI Model Research Agent Flow

Version: 2.0.0 **Last Updated:** 2025-01-09 **Status:** Architecture Design **Document Type:** Comprehensive Technical Specification

---

Table of Contents

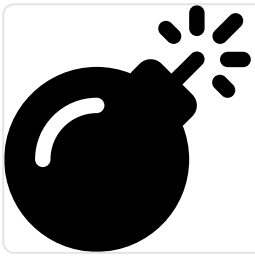
Overview

Executive Summary

The **ROI Model Research Agent Flow** represents a fundamental architectural shift from qualitative value proposition analysis to **quantitative, model-driven ROI calculations**. This system enables healthcare organizations to create, refine, and deploy sophisticated ROI models that drive data-backed dashboard visualizations for prospect engagement.

Key Architectural Changes

This redesign transforms how the system processes client information and generates insights:



Syntax error in text

mermaid version 11.12.1

Concept	Old System	New System	Impact
Primary Entity	Value Proposition (qualitative text)	**ROI Model** (quantitative structure)	Enables precise calculations
Input Document	Client brief/ documents	**ROI Story** (PDF/ Markdown/Bullets/ SQL)	Flexible input formats
Classification	None	**ROI Model Type** (1 of 13 types)	Structured categorization
Variables	Hardcoded in templates	**Dynamic variables** extracted from ROI Story	Customizable per client
Formulas	None	**Mathematical formulas** for ROI calculations	Real ROI computation
Output	Static dashboard templates	**ROI Model** → Data-driven dashboard templates	Live data integration
Validation	Template structure only	**4-layer validation** (JSON, Pydantic, Business, Math)	Robust quality control

System Capabilities

What the System Can Do:

1. **Intelligent Classification**: Automatically categorize ROI Stories into 13 distinct model types
2. **Model Construction**: Build complete ROI models with variables, formulas, and assumptions
3. **Interactive Refinement**: Allow users to adjust models through direct edits or natural language prompts
4. **Dashboard Generation**: Create 5-10 dashboard variations optimized for different audiences
5. **Data Integration**: Execute ROI calculations on real prospect data via MARÉ Runtime
6. **Multi-Format Support**: Accept ROI Stories as PDF, Markdown, bullet points, or SQL queries

Business Value:

- **For Clients**: Quantify ROI with mathematical precision, not estimates
- **For Prospects**: Transparent calculations backed by their own data
- **For Sales Teams**: Multiple dashboard views for different stakeholders
- **For Operations**: Reusable ROI Model templates across prospects

---

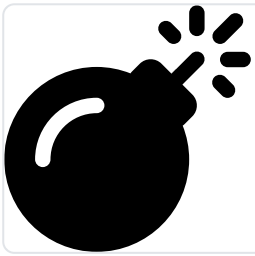
13 ROI Model Types

Overview

The system classifies all ROI Stories into one of **13 distinct ROI Model Types**, each representing a different healthcare cost reduction or value creation strategy. Each type has its own:

- **Variable templates** (baseline costs, targets, intervention parameters)
- **Formula structures** (ROI calculations, payback periods, savings projections)
- **Data requirements** (claims types, member data, clinical outcomes)
- **Industry benchmarks** (typical ranges for metrics)

Type Classification Matrix



Syntax error in text

mermaid version 11.12.1

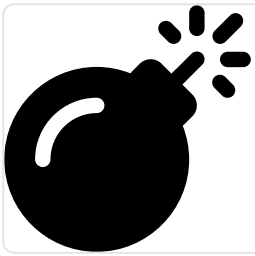
Detailed Type Specifications

#	ROI Model Type	Category	Key Metrics	Typical ROI Range	Data Requirements
1	**Unit Price Reduction**	Cost Reduction	Unit cost PMPM, contract savings %	150-300%	Claims with price/ negotiation data
2	**Site of Care Shift**	Utilization	HOPD vs freestanding cost delta, shift %	200-400%	Facility claims, place of service
3	**Provider Steering**	Cost Reduction	Tier differential, steering rate	180-350%	Provider network data, quality scores
4	**Payment Integrity**	Administrative	Overpayment recovery $, edit hit rate	500-1000%	Claims + contract rules
5	**Utilization Reduction**	Utilization	Low-value service rate, reduction %	200-450%	Utilization patterns, clinical guidelines
6	**Medical Management**	Utilization	Chronic disease costs, outcome improvement	250-500%	Clinical data, member health status
7	**Episode Optimization**	Utilization	Episode cost, pathway adherence %	150-300%	Episode grouper data, care pathways
8	**OON Mitigation**	Administrative	OON rate, cost differential	300-600%	OON claims, negotiation outcomes

#	ROI Model Type	Category	Key Metrics	Typical ROI Range	Data Requirements
9	**Leakage Recapture**	Utilization	Leakage %, in-system cost advantage	200-400%	Referral patterns, system attribution
10	**Pharmacy Optimization**	Cost Reduction	Pharmacy PMPM, specialty drug %, adherence	250-400%	Pharmacy claims, NDC data, adherence
11	**Supply Chain Validation**	Administrative	Device/ implant cost variance	400-800%	Supply claims, UDI data, invoices
12	**Admin Error Reduction**	Administrative	Duplicate payment $, eligibility errors	600-1200%	Claims processing data, member eligibility
13	**Vendor Incentive ROI**	Utilization	Incentive $, steerage impact	100-250%	Vendor program data, behavior change metrics

Type Selection Logic

The ROI Story Classification Agent uses this decision tree to determine the appropriate type:



Syntax error in text

mermaid version 11.12.1

Example: Type 10 - Pharmacy Optimization

Detailed Specification:

Description: Optimize pharmacy spending through specialty drug management, dose optimization, adherence programs, and formulary management.

Key Variables:

```
{
  "population": {
    "total_members": "integer",
    "specialty_drug_users": "integer",
    "specialty_drug_percent": "float (0-1)"
  },
  "baseline_costs": {
    "total_pharmacy_pmpm": "currency",
    "specialty_pharmacy_pmpm": "currency",
    "generic_rate": "float (0-1)"
  },
  "intervention_parameters": {
    "dose_optimization_savings_percent": "float (0-1)",
    "adherence_improvement_percent": "float (0-1)",
    "formulary_savings_percent": "float (0-1)",
    "step_therapy_impact": "float (0-1)"
  },
  "targets": {
    "target_specialty_pmpm": "currency",
    "target_adherence_rate": "float (0-1)",
    "target_generic_rate": "float (0-1)"
  }
}
```

Standard Formulas:

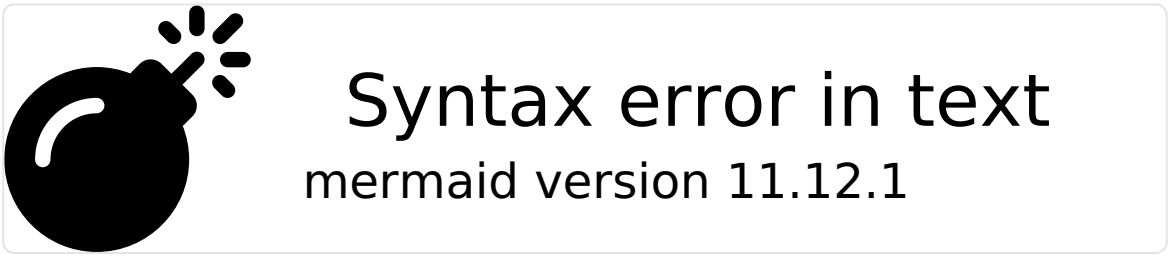
```
{
  "annual_baseline_specialty_cost": "specialty_pharmacy_pmpm * 12 * spec
  "dose_optimization_savings": "annual_baseline_specialty_cost * dose_op
  "adherence_savings": "annual_baseline_specialty_cost * adherence_impro
  "formulary_savings": "annual_baseline_specialty_cost * formulary_savin
  "total_annual_savings": "dose_optimization_savings + adherence_savings
  "roi_percent": "(total_annual_savings / intervention_cost) * 100",
  "payback_months": "intervention_cost / (total_annual_savings / 12)",
  "savings_pmpm": "total_annual_savings / (total_members * 12)"
}
```

Data Sources Required: - `pharmacy_claims` (NDC-level detail) - `member_eligibility` (enrollment continuity) - `drug_list_with_ndc` (specialty flagging) - `specialty_classification` (specialty drug taxonomy) - `adherence_data` (PDC scores if available)

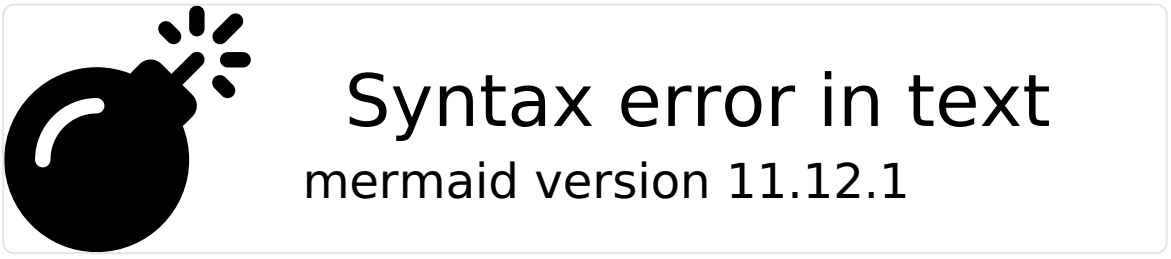
Industry Benchmarks: - Specialty drug spend: 40-50% of total pharmacy - Dose optimization savings: 10-20% - Adherence improvement impact: 8-15% - Formulary optimization: 5-10% - Combined ROI: 250-400%

System Architecture

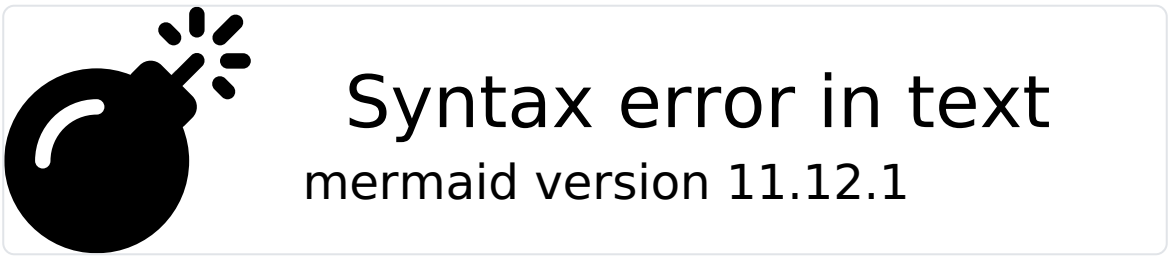
High-Level System Overview



Agent Interaction Architecture

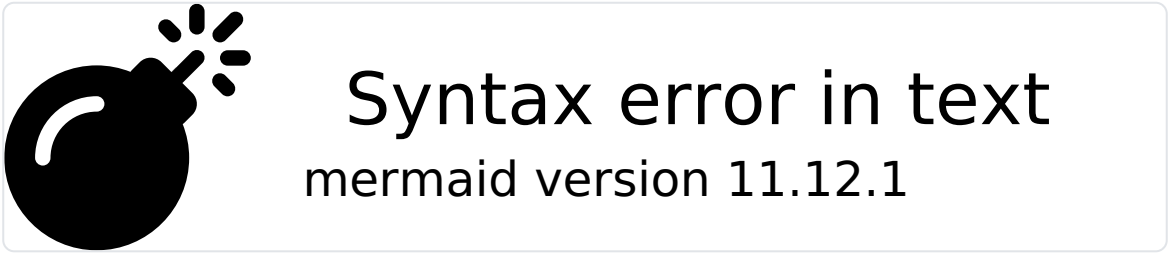


Data Flow Architecture



Agent Ecosystem

Agent Hierarchy & Responsibilities

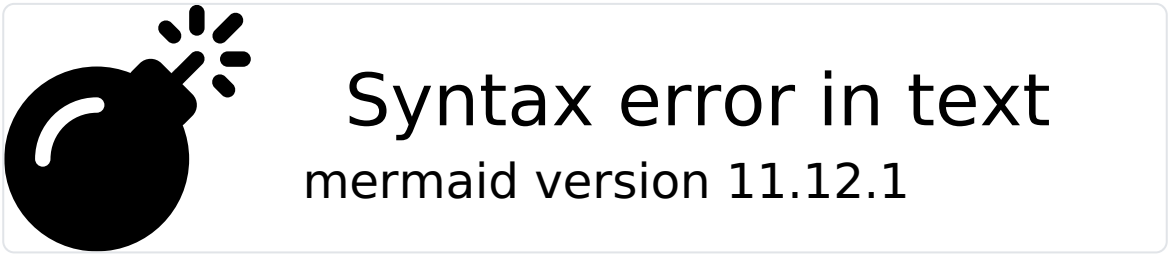


1. ROI Story Classification Agent (NEW)

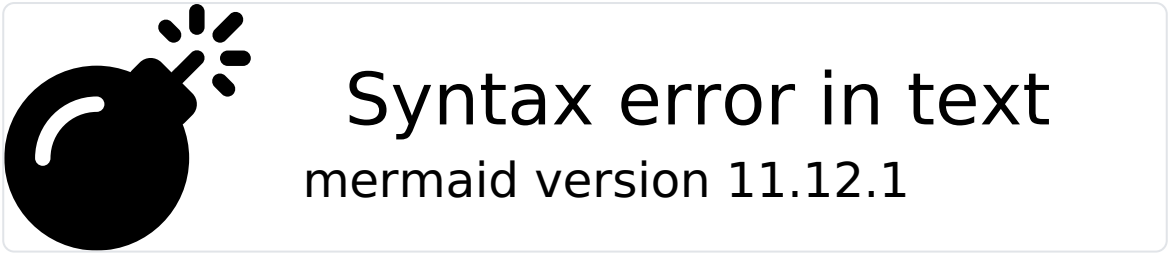
Purpose & Capabilities

The **ROI Story Classification Agent** is the entry point of the ROI Model pipeline. It analyzes uploaded ROI Stories and intelligently categorizes them into one of 13 distinct ROI Model Types, enabling the system to apply the correct variable templates and calculation logic.

Input Processing Flow

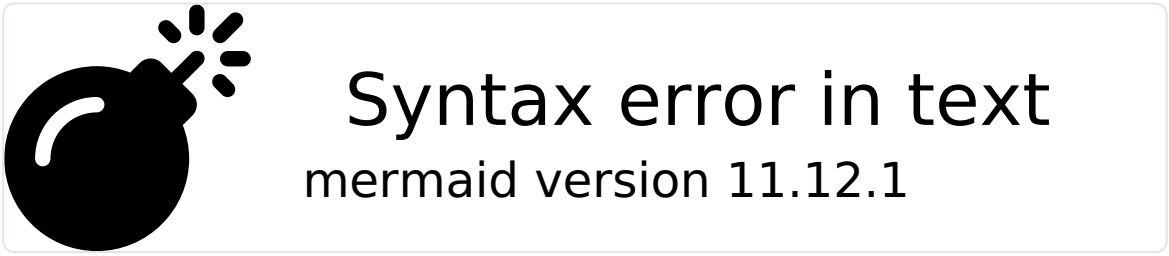


Classification Algorithm



Confidence Scoring Model

The agent calculates confidence based on multiple factors:



Confidence Score Breakdown:

Factor	Weight	Scoring Method
Keyword Matches	30%	Count of type-specific keywords found / total expected keywords
Metric Presence	25%	Presence of expected metrics (PMPM, %, savings $)
Theme Alignment	25%	Semantic similarity to type description (via embeddings)
Structure Match	20%	Document structure matches typical format for type

Agent Configuration

```
 # Agent: ROI Story Classification Agent
{
    "name": "ROIStoryClassificationAgent",
    "model": "aws_bedrock/claude-sonnet-4",
    "temperature": 0.1,  # Low temperature for consistent classification
    "max_tokens": 8000,
    "tools": [],  # No external tools needed
    "instructions_file": "agents/templates/roi_story_classification_inst
    "retry_logic": {
        "enabled": True,
        "max_retries": 3,
        "validation": "one_of_13_types_with_confidence"
    },
    "output_schema": "ROIStoryClassificationResult"
}
```

Input Schema

```
 {
  "roi_story_content": "string (extracted text)",
  "roi_story_type": "pdf | markdown | bullets | sql",
  "client_context": {
    "industry": "health_plan | employer | provider | tpa",
    "member_population": "integer (optional)",
    "focus_areas": ["array of focus hints (optional)"]
  }
}
```

Output Schema

```
{
 "primary_roi_model_type": "Pharmacy Optimization",
 "primary_type_id": 10,
 "confidence": 0.92,

 "secondary_types": [
   {
     "type": "Utilization Reduction",
     "type_id": 5,
     "confidence": 0.68,
     "reasoning": "Secondary themes of reducing unnecessary utilization
   }
 ],

 "reasoning": "ROI Story focuses heavily on specialty drug cost reducti

 "key_indicators_found": [
   "specialty drug costs",
   "dose optimization",
   "adherence programs",
   "PMPM pharmacy spend",
   "formulary management",
   "specialty drug percentage"
 ],

 "missing_indicators": [
   "intervention cost (not specified)",
   "current adherence rate (assumed)"
 ],

 "suggested_variables": [
   "baseline_pmpm_rx",
   "specialty_drug_percent",
   "adherence_rate_baseline",
   "target_adherence_rate",
   "dose_optimization_savings_percent",
   "formulary_savings_percent"
```

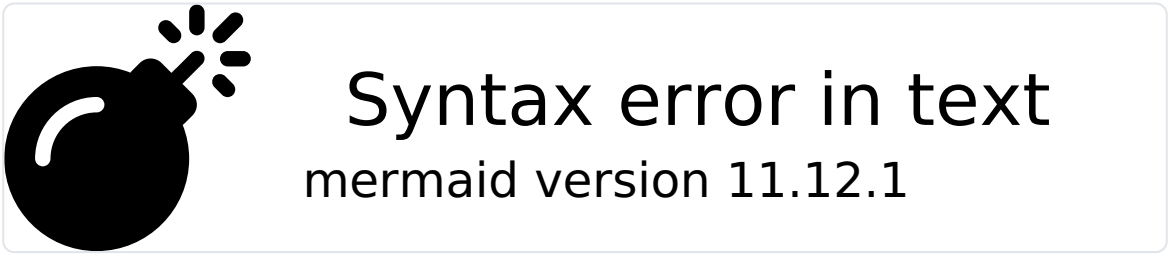
```
  ],

  "extracted_metrics": {
    "baseline_pharmacy_pmpm": 120.00,
    "specialty_pharmacy_pmpm": 85.00,
    "target_savings_percent": 0.20,
    "population_size": 50000
  },

  "classification_metadata": {
    "processing_time_seconds": 12.4,
    "retry_attempts": 0,
    "model_version": "claude-sonnet-4-20250514",
    "classification_timestamp": "2025-01-09T14:32:10Z"
  }
}
```

Validation Rules

The Classification Agent must pass these validations:

1. **Type Validation**: `primary_type_id` must be between 1-13
2. **Confidence Threshold**: `confidence` >= 0.5 (otherwise flagged for manual review)
3. **Reasoning Required**: `reasoning` field must be >= 50 characters
4. **Indicators Found**: At least 2 key indicators must be identified
5. **Suggested Variables**: At least 3 variables must be suggested

Error Handling & Retry Logic



---

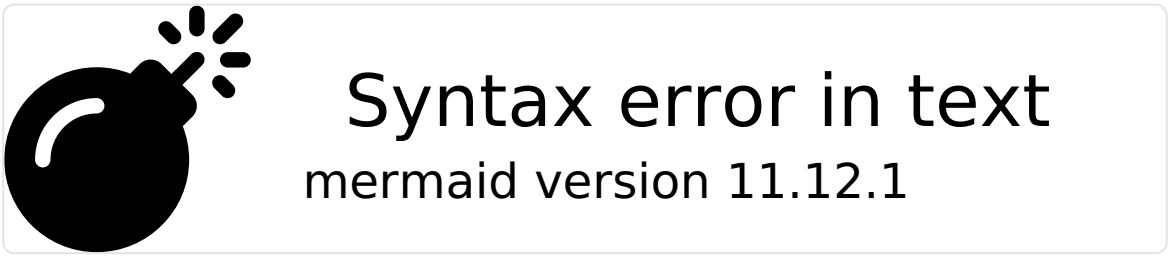
2. ROI Model Builder Agent (NEW)

Purpose & Capabilities

The **ROI Model Builder Agent** constructs complete, mathematically-valid ROI Models from the classified ROI Story. It is the most complex agent in the system, responsible for:

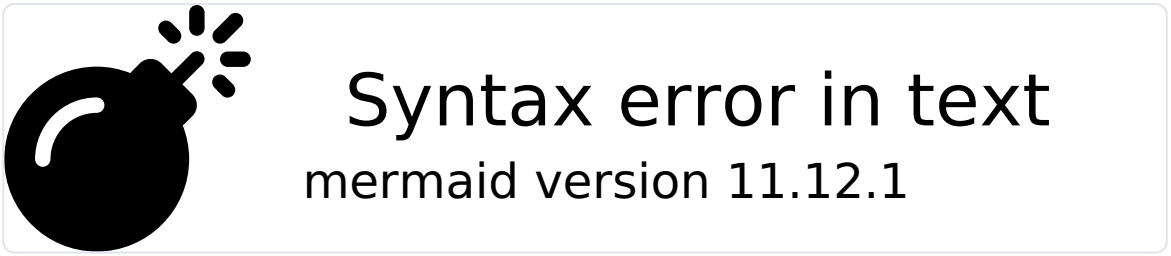
- Loading type-specific variable templates
- Extracting values from ROI Story content
- Populating variables with extracted or assumed values
- Defining formulas with correct variable references
- Calculating derived metrics
- Validating mathematical consistency (no circular dependencies)
- Identifying data source requirements

Model Building Process

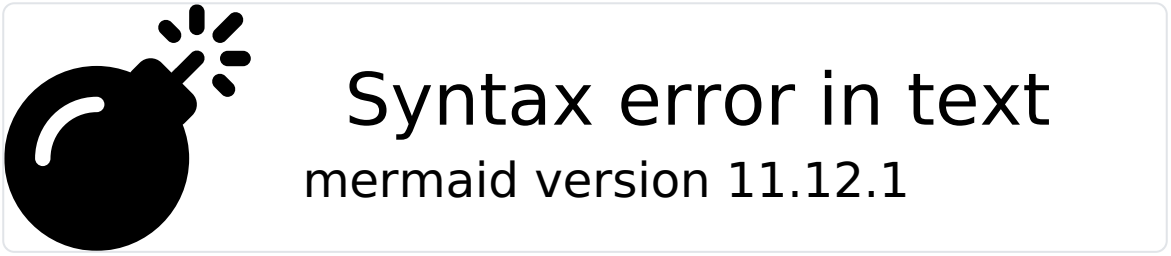


Template Loading Logic

Each of the 13 ROI Model Types has a predefined template:



Value Extraction Strategy



Variable Confidence Scoring

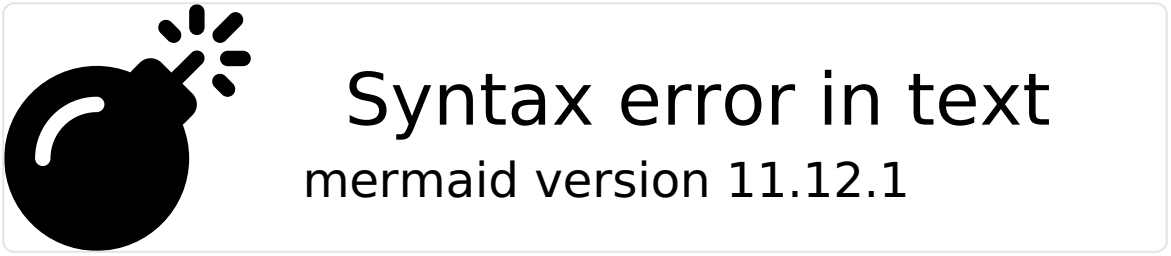
Each variable gets a confidence score based on its source:

Source	Confidence	Notes
Extracted from ROI Story (explicit)	1.0	Value directly stated in document
Extracted (inferred)	0.8	Value calculated from other explicit values
Industry Benchmark	0.6	Standard industry average applied
Calculated Default	0.5	Derived from related variables
User Input Required	0.0	Placeholder - needs user input

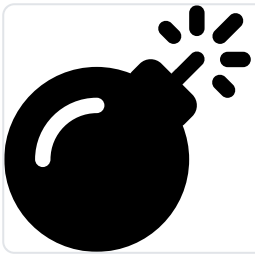
Overall model confidence = weighted average of all variable confidences.

Formula Dependency Graph

The agent builds a dependency graph to ensure no circular references:



Circular Dependency Detection:



Syntax error in text

mermaid version 11.12.1

Agent Configuration

```
 # Agent: ROI Model Builder Agent
{
    "name": "ROIModelBuilderAgent",
    "model": "aws_bedrock/claude-sonnet-4",
    "temperature": 0.0,  # Deterministic for math calculations
    "max_tokens": 16000,
    "tools": ["roi_model_calculator"],
    "instructions_file": "agents/templates/roi_model_builder_instruction
    "type_specific_templates": "agents/templates/roi_model_builders/",
    "retry_logic": {
        "enabled": True,
        "max_retries": 3,
        "validation": "complete_model_with_valid_formulas"
    },
    "output_schema": "ROIModelBuilderResult"
}
```

Input Schema


```
{
  "roi_model_type_id": 10,
  "roi_model_type_name": "Pharmacy Optimization",
  "roi_story_content": "Full text of ROI Story",
  "extracted_metrics": {
    "baseline_pharmacy_pmpm": 120.00,
    "specialty_pharmacy_pmpm": 85.00,
    "target_savings_percent": 0.20,
    "population_size": 50000
  },
  "client_context": {
    "industry": "health_plan",
    "member_population": 50000,
    "geography": "national",
    "additional_context": "Focus on specialty drug management"
  }
}
```

Output Schema (Detailed Example)

```
{
 "roi_model_id": "a7f3c8e2-4b1d-4a9e-8f2c-1d6e9b3a5c8f",
 "roi_model_name": "Specialty Pharmacy Optimization - ABC Health Plan",
 "roi_model_type": "Pharmacy Optimization",
 "roi_model_type_id": 10,

 "variables": {
   "population": {
     "total_members": {
       "value": 50000,
       "unit": "members",
       "source": "extracted",
       "confidence": 1.0,
       "editable": true
     },
     "specialty_drug_users": {
       "value": 2500,
       "unit": "members",
       "source": "calculated",
       "calculation": "total_members * specialty_drug_percent",
       "confidence": 0.8,
       "editable": false
     },
     "specialty_drug_percent": {
       "value": 0.05,
       "unit": "percent",
       "source": "industry_benchmark",
       "confidence": 0.6,
       "editable": true,
       "benchmark_range": "0.03-0.07"
     }
   },

   "baseline_costs": {
     "total_medical_pmpm": {
       "value": 450.00,
       "unit": "currency_pmpm",
```

```
        "source": "extracted",
        "confidence": 1.0,
        "editable": true
      },
      "pharmacy_pmpm": {
        "value": 120.00,
        "unit": "currency_pmpm",
        "source": "extracted",
        "confidence": 1.0,
        "editable": true
      },
      "specialty_pharmacy_pmpm": {
        "value": 85.00,
        "unit": "currency_pmpm",
        "source": "extracted",
        "confidence": 1.0,
        "editable": true
      },
      "generic_rate": {
        "value": 0.82,
        "unit": "percent",
        "source": "industry_benchmark",
        "confidence": 0.6,
        "editable": true,
        "benchmark_range": "0.75-0.90"
      }
    },

    "intervention_parameters": {
      "intervention_cost": {
        "value": 4200000.00,
        "unit": "currency",
        "source": "user_required",
        "confidence": 0.0,
        "editable": true,
        "required": true,
        "prompt": "Enter the total intervention cost (technology, implem
```

```
    },
    "dose_optimization_savings_percent": {
      "value": 0.15,
      "unit": "percent",
      "source": "industry_benchmark",
      "confidence": 0.6,
      "editable": true,
      "benchmark_range": "0.10-0.20"
    },
    "adherence_improvement_percent": {
      "value": 0.12,
      "unit": "percent",
      "source": "industry_benchmark",
      "confidence": 0.6,
      "editable": true,
      "benchmark_range": "0.08-0.15"
    },
    "formulary_savings_percent": {
      "value": 0.08,
      "unit": "percent",
      "source": "industry_benchmark",
      "confidence": 0.6,
      "editable": true,
      "benchmark_range": "0.05-0.10"
    }
  },

  "targets": {
    "target_specialty_pmpm": {
      "value": 68.00,
      "unit": "currency_pmpm",
      "source": "calculated",
      "calculation": "specialty_pharmacy_pmpm * (1 - target_savings_pe
      "confidence": 0.8,
      "editable": true
    },
    "target_savings_pmpm": {
```

```
        "value": 17.00,
        "unit": "currency_pmpm",
        "source": "calculated",
        "calculation": "specialty_pharmacy_pmpm - target_specialty_pmpm"
        "confidence": 0.8,
        "editable": false
      },
      "implementation_months": {
        "value": 12,
        "unit": "months",
        "source": "industry_benchmark",
        "confidence": 0.6,
        "editable": true,
        "benchmark_range": "6-18"
      },
      "full_run_rate_months": {
        "value": 18,
        "unit": "months",
        "source": "industry_benchmark",
        "confidence": 0.6,
        "editable": true,
        "benchmark_range": "12-24"
      }
    }
  },

  "formulas": {
    "annual_baseline_specialty_cost": {
      "formula": "specialty_pharmacy_pmpm * 12 * specialty_drug_users",
      "result": 2550000.00,
      "unit": "currency",
      "dependencies": ["specialty_pharmacy_pmpm", "specialty_drug_users"
    },
    "annual_target_specialty_cost": {
      "formula": "target_specialty_pmpm * 12 * specialty_drug_users",
      "result": 2040000.00,
      "unit": "currency",
```

```
      "dependencies": ["target_specialty_pmpm", "specialty_drug_users"]
    },
    "annual_savings": {
      "formula": "annual_baseline_specialty_cost - annual_target_special
      "result": 510000.00,
      "unit": "currency",
      "dependencies": ["annual_baseline_specialty_cost", "annual_target_
    },
    "dose_optimization_savings": {
      "formula": "annual_baseline_specialty_cost * dose_optimization_sav
      "result": 382500.00,
      "unit": "currency",
      "dependencies": ["annual_baseline_specialty_cost", "dose_optimizat
    },
    "adherence_savings": {
      "formula": "annual_baseline_specialty_cost * adherence_improvement
      "result": 306000.00,
      "unit": "currency",
      "dependencies": ["annual_baseline_specialty_cost", "adherence_impr
    },
    "formulary_savings": {
      "formula": "annual_baseline_specialty_cost * formulary_savings_per
      "result": 204000.00,
      "unit": "currency",
      "dependencies": ["annual_baseline_specialty_cost", "formulary_savi
    },
    "roi_percent": {
      "formula": "(annual_savings / intervention_cost) * 100",
      "result": null,
      "unit": "percent",
      "dependencies": ["annual_savings", "intervention_cost"],
      "note": "Cannot calculate - intervention_cost not provided"
    },
    "payback_months": {
      "formula": "intervention_cost / (annual_savings / 12)",
      "result": null,
      "unit": "months",
```

```
      "dependencies": ["intervention_cost", "annual_savings"],
      "note": "Cannot calculate - intervention_cost not provided"
    },
    "savings_pmpm": {
      "formula": "annual_savings / (total_members * 12)",
      "result": 0.85,
      "unit": "currency_pmpm",
      "dependencies": ["annual_savings", "total_members"]
    }
  },

  "assumptions": [
    "Specialty drug users remain constant at 5% of population throughout
    "Baseline PMPM costs include current adherence rate of approximately
    "Dose optimization applies to 40% of specialty drug users",
    "Full savings realized by month 18 with linear ramp-up starting mont
    "No significant change in specialty drug mix or population health st
    "Generic rate remains stable at 82% throughout period",
    "Intervention cost not specified - user input required for ROI calcu
  ],

  "data_sources_needed": [
    {
      "source": "pharmacy_claims",
      "fields": ["ndc", "fill_date", "days_supply", "quantity", "drug_co
      "purpose": "Calculate baseline PMPM costs and specialty drug utili
    },
    {
      "source": "member_eligibility",
      "fields": ["member_id", "effective_date", "termination_date", "cov
      "purpose": "Determine eligible member months for PMPM calculations
    },
    {
      "source": "drug_list_with_ndc",
      "fields": ["ndc", "drug_name", "specialty_flag", "therapeutic_clas
      "purpose": "Classify drugs as specialty vs generic for segmentatio
    },
```

```
    {
      "source": "specialty_classification",
      "fields": ["ndc", "specialty_category", "dose_optimization_eligibl
      "purpose": "Identify opportunities for dose optimization"
    }
  ],

  "validation_results": {
    "all_required_variables_present": false,
    "missing_required_variables": ["intervention_parameters.intervention
    "formula_dependencies_valid": true,
    "circular_dependencies": false,
    "math_checks_passed": true,
    "unreasonable_values": []
  },

  "confidence": 0.75,
  "confidence_breakdown": {
    "variable_extraction": 0.85,
    "formula_validity": 1.0,
    "assumptions_quality": 0.70,
    "completeness": 0.65
  },

  "missing_information": [
    "Intervention cost not specified in ROI Story - required for ROI and
    "Current adherence rate assumed at 65% based on industry average - a
    "Dose optimization eligibility percentage assumed at 40% - should be
  ],

  "warnings": [
    "Target savings of 20% is on the higher end of industry benchmarks (
    "Specialty drug percentage of 5% may vary significantly by populatio
  ],

  "next_steps": [
    "User should provide intervention cost estimate",
```

```
    "Review and adjust industry benchmark assumptions",
    "Validate specialty drug percentage against actual population data",
    "Consider adding multi-year projection scenarios"
  ],

  "metadata": {
    "created_at": "2025-01-09T14:45:22Z",
    "model_version": "claude-sonnet-4-20250514",
    "build_duration_seconds": 34.2,
    "retry_attempts": 0,
    "template_version": "pharmacy_optimization_v2.1"
  }
}
```

Validation Rules

1. **Variable Completeness**: All required variables present (even if confidence = 0.0 for user input)
2. **Formula Validity**: All formulas reference existing variables
3. **No Circular Dependencies**: Topological sort succeeds on dependency graph
4. **Math Consistency**: No divide-by-zero, negative values where impossible
5. **Unit Consistency**: Operations on compatible units (can't add currency to percent)
6. **Range Validation**: Values within reasonable ranges for healthcare (e.g., PMPM < $10,000)

---

3. ROI Model Refinement Agent (NEW)

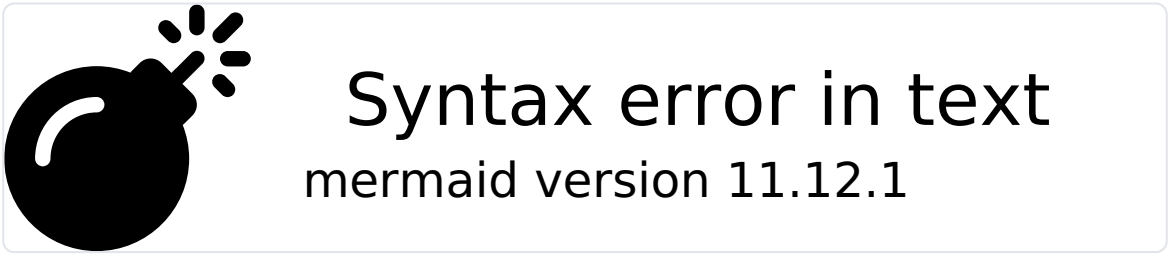
Purpose & Capabilities

The **ROI Model Refinement Agent** enables iterative improvement of ROI Models through:

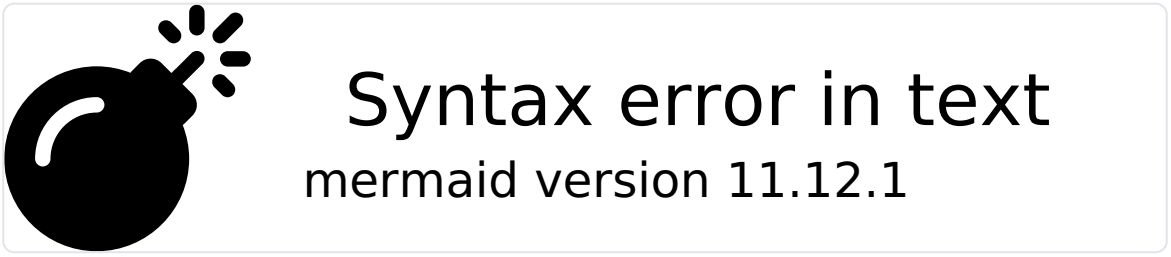
- **Direct Variable Updates**: User changes specific values
- **Natural Language Prompts**: User requests modifications via text (e.g., "Increase savings target to 25%")
- **Scenario Modeling**: Add multi-year projections, sensitivity analysis

- **Formula Adjustments**: Modify calculation logic per user needs

Refinement Flow



Natural Language Parsing

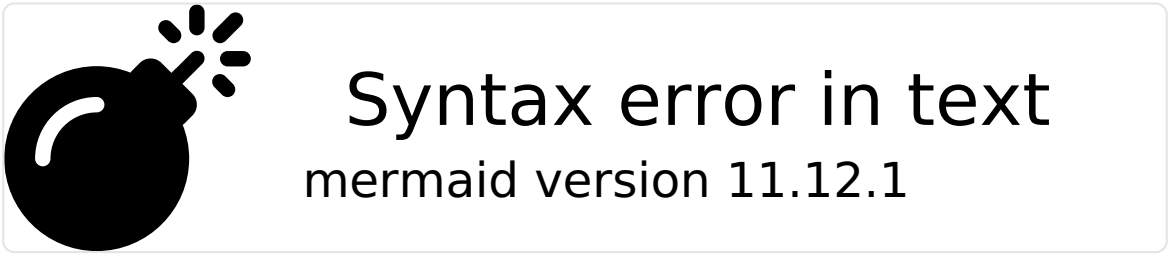


Example Prompt Parsing:

User Prompt	Parsed Intent	Extracted Values
"Increase target savings to 25%"	`update_target`	`target_savings_percent: 0.25`
"Add 3-year projections"	`add_scenario`	`years: 3, scenario_type: multi_year`
"Change intervention cost to $5M"	`update_variable`	`intervention_cost: 5000000.00`
"Remove adherence improvement assumption"	`remove_assumption`	`assumption_id: adherence_improvement`
"Use actual population of 62,000 members"	`update_variable`	`total_members: 62000`

Dependency Recalculation

When a variable changes, the agent must recalculate all dependent formulas:



Change Tracking

Every refinement creates a detailed change log:

```
{
 "refinement_id": "ref_abc123",
 "refinement_timestamp": "2025-01-09T15:30:45Z",
 "user_prompt": "Increase target savings to 25% and add 3-year projecti

 "changes_made": [
   {
     "change_type": "variable_update",
     "field": "variables.targets.target_savings_percent",
     "old_value": 0.20,
     "new_value": 0.25,
     "reason": "User requested increase to 25%",
     "confidence_change": {"before": 0.75, "after": 0.80}
   },
   {
     "change_type": "formula_recalculation",
     "field": "formulas.annual_savings",
     "old_result": 510000.00,
     "new_result": 637500.00,
     "reason": "Recalculated due to target_savings_percent change"
   },
   {
     "change_type": "formula_recalculation",
     "field": "formulas.roi_percent",
     "old_result": null,
     "new_result": null,
     "reason": "Cannot calculate - intervention_cost still missing"
   },
   {
     "change_type": "scenario_addition",
     "field": "scenarios.three_year_projection",
     "new_value": {
       "year_1_savings": 637500.00,
       "year_2_savings": 701250.00,
       "year_3_savings": 771375.00,
       "cumulative_savings": 2110125.00,
       "assumptions": ["10% year-over-year improvement"]
```

```
      },
      "reason": "User requested 3-year projections"
    }
  ],

  "validation_status": "valid",

  "warnings": [
    "Target savings of 25% exceeds industry benchmark range (15-20%)",
    "3-year projection assumes consistent 10% improvement - may not be r
    "Cumulative savings calculation does not account for discount rate"
  ],

  "recommendations": [
    "Consider adding sensitivity analysis for 20%, 22.5%, 25% savings sc
    "Apply discount rate to multi-year projections for NPV calculation",
    "Validate 25% target with clinical team before presenting to prospec
  ]
}
```

Agent Configuration

```
 # Agent: ROI Model Refinement Agent
{
    "name": "ROIModelRefinementAgent",
    "model": "aws_bedrock/claude-sonnet-4",
    "temperature": 0.2,  # Slightly creative for scenario generation
    "max_tokens": 12000,
    "tools": ["roi_model_calculator"],
    "instructions_file": "agents/templates/roi_model_refinement_instruct
    "retry_logic": {
        "enabled": True,
        "max_retries": 3,
        "validation": "valid_updated_model_with_change_log"
    },
    "output_schema": "ROIModelRefinementResult"
}
```

---

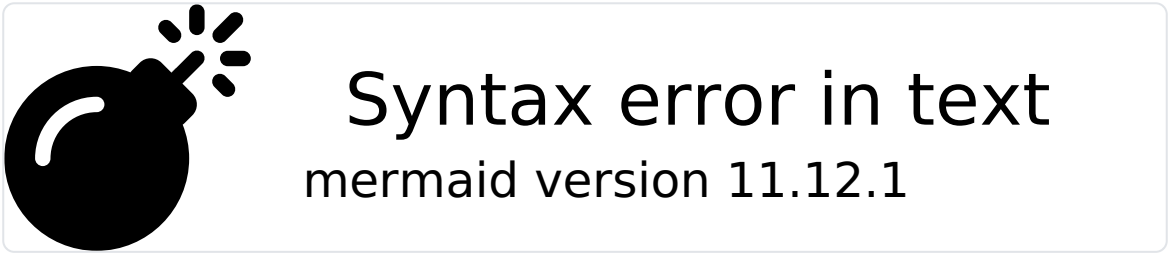
4. Dashboard Template Generator Agent (UPDATED)

Purpose & Capabilities (Updated for ROI Models)

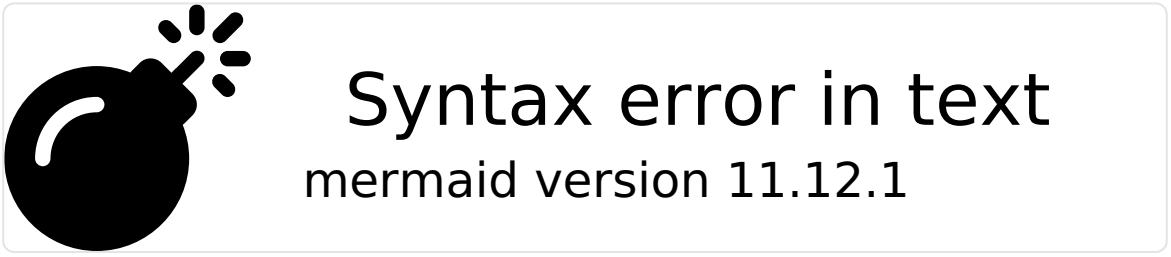
The **Dashboard Template Generator Agent** has been significantly updated to work with **ROI Models** instead of qualitative value propositions. Key changes:

- **Input**: ROI Model (variables, formulas, assumptions) instead of text-based value proposition
- **Widget Data Sources**: Directly reference ROI Model formula results
- **Chart Types**: Match to variable types (currency → bar/line, percent → gauge, trend → area)
- **Multi-Audience**: Generate audience-specific views from same ROI Model

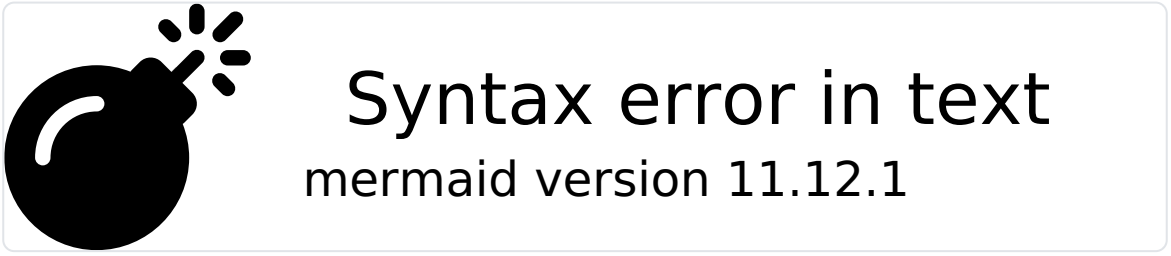
Dashboard Generation Process



Widget Type Selection Logic



Audience-Specific Dashboard Variations



Example: Widget Configuration for ROI Model

ROI Model Formula:

```
{
  "formulas": {
    "annual_savings": {
      "formula": "annual_baseline_specialty_cost - annual_target_special
      "result": 510000.00,
      "unit": "currency"
    }
  }
}
```


Generated Widget:


```
{
  "widget_id": "w_annual_savings",
  "title": "Annual Savings",
  "type": "metric",
  "chartType": "kpi",
  "config": {
    "data_source": "roi_model.formulas.annual_savings.result",
    "format": "currency",
    "display_format": "$510,000",
    "comparison_baseline": "roi_model.formulas.annual_baseline_specialty
    "comparison_type": "percentage_of_baseline",
    "icon": "dollar-sign",
    "color": "green",
    "tooltip": "Total annual savings from specialty pharmacy optimizatio
  },
  "position": {
    "row": 1,
    "col": 4,
    "rowSpan": 1,
    "colSpan": 3
  }
}
```

Dashboard Template Categories

Each ROI Model generates **5-10 dashboard templates** across these categories:

Category	Description	Typical Widgets	Target Audience
Executive Summary	High-level KPIs and ROI	4-6 KPI cards, 1-2 summary charts	C-suite, Executives
Financial Deep Dive	Detailed cost breakdown	Waterfall, cost trends, breakdown tables	CFO, Finance teams
Clinical Impact	Health outcomes and quality	Outcome metrics, quality measures	CMO, Clinical teams
Implementation Timeline	Project milestones and progress	Gantt chart, milestone tracker	Project managers
Savings Waterfall	Breakdown of savings sources	Waterfall chart, category breakdown	Finance, Operations
Population Analysis	Member/patient demographics and impact	Demographics, utilization patterns	Actuaries, Analysts
Audience-Specific	Tailored to Health Plan, Employer, Provider	Varies by audience	Specific stakeholder groups

---

5. Web Search Agent (EXISTING - Context Updated)

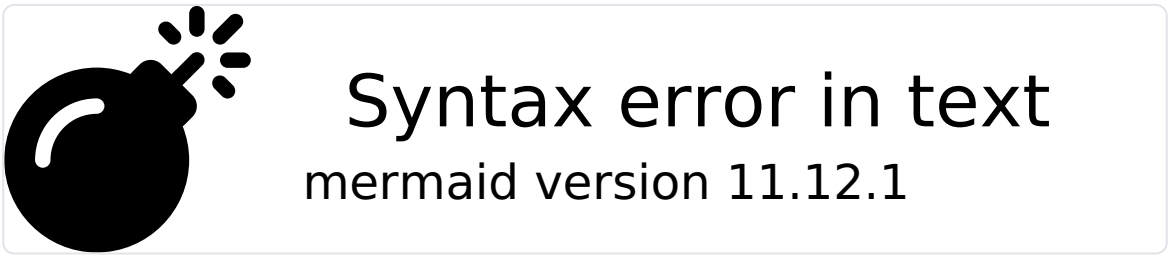
Updated Usage in ROI Model Flow

The **Web Search Agent** continues to serve its original purpose (company research) but now has **additional use cases** in the ROI Model flow:

New Use Cases:

1. **ROI Story Enrichment**: When ROI Story is minimal (e.g., "Reduce pharmacy costs"), search for:
    - Industry benchmarks for pharmacy optimization ROI
    - Case studies with quantitative results
    - Typical intervention costs and timelines
    - Best practices and savings ranges
2. **Industry Benchmark Validation**: Validate assumed variables against public data:
    - "What is typical specialty drug percentage in health plans?"
    - "Average adherence rate for chronic medications"
    - "Pharmacy cost reduction case studies"
3. **Competitive Intelligence**: Research prospect's competitors:
    - Similar programs they've implemented
    - Public ROI claims
    - Market positioning

Integration with ROI Model Pipeline



---

6. Document Analysis Agent (EXISTING - Context Updated)

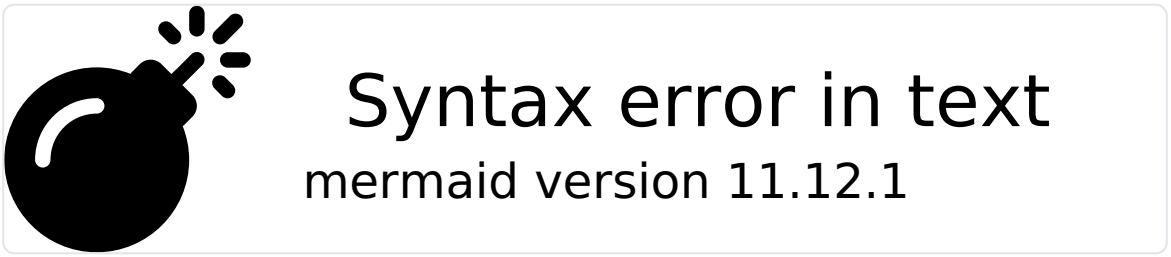
Updated Usage in ROI Model Flow

The **Document Analysis Agent** extracts structured data from uploaded ROI Story documents, now with **enhanced extraction patterns** for ROI-specific metrics.

Enhanced Extraction Targets:

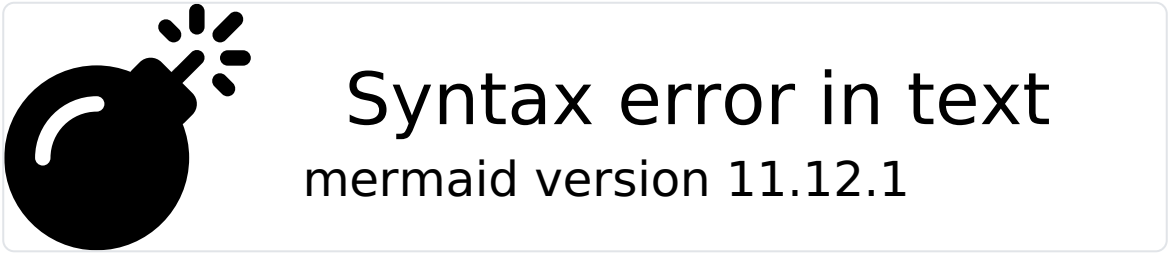
Document Section	Extraction Pattern	Variable Mapping
Baseline Costs	"\$XXX PMPM", "PMPM: $XXX"	`baseline_pharmacy_pmpm`
Target Savings	"XX% reduction", "save XX%"	`target_savings_percent`
Population	"XX,XXX members", "population: XXK"	`total_members`
Intervention Cost	"$X.XM investment", "cost: $XXM"	`intervention_cost`
Timeline	"XX month implementation", "XX months to full run-rate"	`implementation_months`, `full_run_rate_months`
ROI Claims	"XXX% ROI", "ROI of XX%"	`expected_roi_percent` (for validation)

Enhanced Extraction Flow

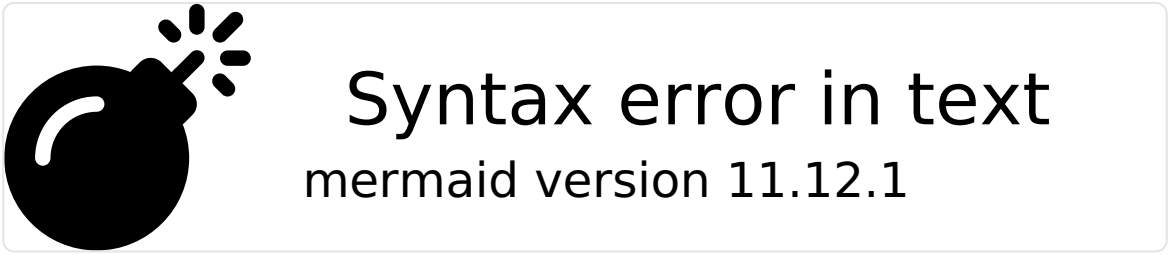


Complete Workflows

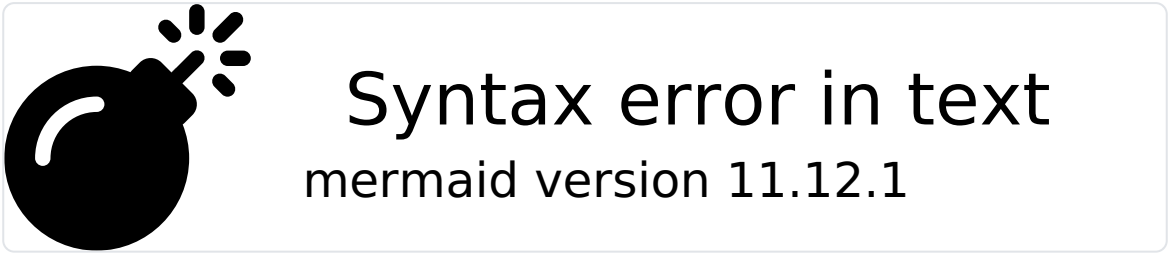
Workflow 1: ROI Model Creation (Client Admin)



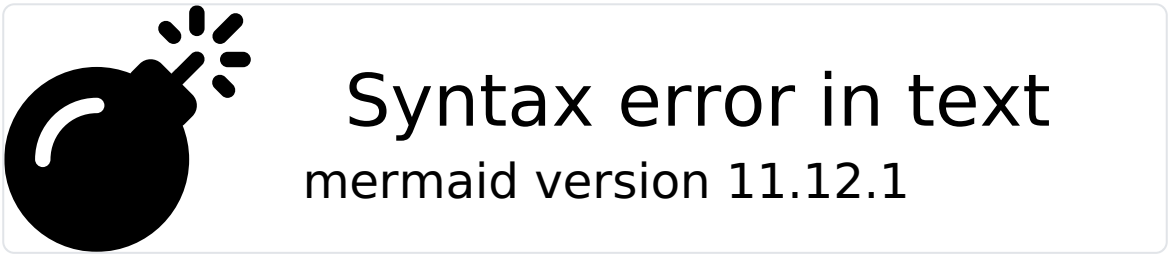
Workflow 2: Dashboard Template Generation



Workflow 3: Prospect Association & Data Upload

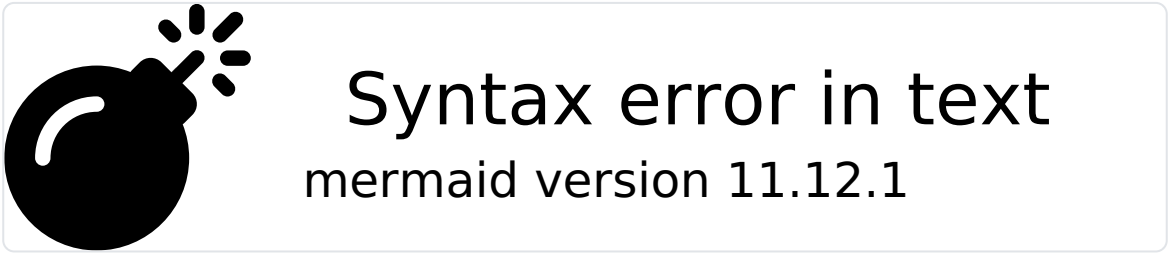


Workflow 4: Error Handling & Retry Flow



API Integration

Complete API Endpoint Map



API Request/Response Flows

1. Create ROI Model → Classification → Build

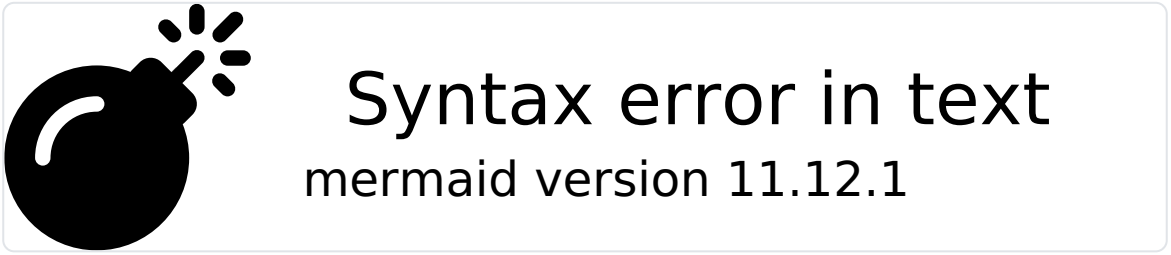
```
 POST /clients/abc-123/roi-models
Content-Type: multipart/form-data

------WebKitFormBoundary
Content-Disposition: form-data; name="roi_model_name"

Specialty Pharmacy Optimization
------WebKitFormBoundary
Content-Disposition: form-data; name="roi_story_file"; filename="pharmac
Content-Type: application/pdf

[Binary PDF content]
------WebKitFormBoundary
Content-Disposition: form-data; name="additional_context"

Health plan with 50K commercial members, focus on specialty drug cost re
------WebKitFormBoundary--

Response (202 Accepted):
{
  "job_id": "roi_classification_7f3a9b2c",
  "roi_model_id": "roi_a7f3c8e2-4b1d-4a9e-8f2c-1d6e9b3a5c8f",
  "status": "pending",
  "message": "ROI Story classification initiated",
  "estimated_completion_seconds": 30,
  "status_url": "/agents/roi-story-classification/roi_classification_7f3
}
```

```
 GET /agents/roi-story-classification/roi_classification_7f3a9b2c

Response (200 OK):
{
  "job_id": "roi_classification_7f3a9b2c",
  "status": "completed",
  "completed_at": "2025-01-09T14:32:10Z",
  "processing_time_seconds": 12.4,
  "result": {
    "primary_roi_model_type": "Pharmacy Optimization",
    "primary_type_id": 10,
    "confidence": 0.92,
    "reasoning": "...",
    "suggested_variables": [...],
    "extracted_metrics": {...}
  }
}
```

```
 POST /agents/roi-model-builder
Content-Type: application/json

{
  "roi_model_id": "roi_a7f3c8e2-4b1d-4a9e-8f2c-1d6e9b3a5c8f",
  "roi_model_type_id": 10,
  "roi_story_content": "...",
  "extracted_metrics": {...},
  "client_context": {...}
}

Response (202 Accepted):
{
  "job_id": "roi_build_3c8e2f1d",
  "status": "pending",
  "estimated_completion_seconds": 45,
  "status_url": "/agents/roi-model-builder/roi_build_3c8e2f1d"
}
```

Database Design

Entity Relationship Diagram



Database Schema SQL

```
 -- ============================================================================
-- ROI Model Types Reference Table
-- ============================================================================
CREATE TABLE roi_model_types (
    roi_model_type_id INTEGER PRIMARY KEY,
    type_name VARCHAR(100) NOT NULL UNIQUE,
    description TEXT NOT NULL,
    category VARCHAR(50) NOT NULL, -- 'cost_reduction', 'utilization', '

    -- Template definitions
    typical_variables JSONB NOT NULL DEFAULT '{}',
    typical_formulas JSONB NOT NULL DEFAULT '{}',
    industry_benchmark_data JSONB NOT NULL DEFAULT '{}',

    -- Metadata
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);

-- Populate 13 ROI Model Types
INSERT INTO roi_model_types (roi_model_type_id, type_name, description,
    (1, 'Unit Price Reduction', 'Lower unit price for same service (netw
    (2, 'Site of Care Shift', 'Shift site of care (HOPD → freestanding;
    (3, 'Provider Steering', 'Steer members to higher-value providers/fa
    (4, 'Payment Integrity', 'Payment integrity & contract compliance (e
    (5, 'Utilization Reduction', 'Reduce unnecessary/low-value utilizati
    (6, 'Medical Management', 'Targeted medical management (chronic, com
    (7, 'Episode Optimization', 'Episode optimization (surgical/MSK/mate
    (8, 'OON Mitigation', 'Out-of-network mitigation (wraps, negotiation
    (9, 'Leakage Recapture', 'Leakage recapture (in-system vs out-of-sys
    (10, 'Pharmacy Optimization', 'Pharmacy optimization (unit cost, uti
    (11, 'Supply Chain Validation', 'Device/implant/supply chain validat
    (12, 'Admin Error Reduction', 'Administrative error reduction (dupli
    (13, 'Vendor Incentive ROI', 'Vendor/member incentive ROI (only if $

-- ============================================================================
-- ROI Models Table
```

```
-- ============================================================================
CREATE TABLE roi_models (
    roi_model_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    client_id UUID NOT NULL REFERENCES clients(client_id) ON DELETE CASC
    roi_model_name VARCHAR(200) NOT NULL,

    -- Classification
    roi_model_type_id INTEGER NOT NULL REFERENCES roi_model_types(roi_mo
    classification_confidence DECIMAL(3,2) CHECK (classification_confide

    -- ROI Story
    roi_story_content TEXT,
    roi_story_type VARCHAR(20), -- 'pdf', 'markdown', 'bullets', 'sql'
    roi_story_s3_path VARCHAR(500),

    -- ROI Model Structure
    variables JSONB NOT NULL DEFAULT '{}',
    formulas JSONB NOT NULL DEFAULT '{}',
    assumptions TEXT[] DEFAULT '{}',
    data_sources_needed TEXT[] DEFAULT '{}',

    -- Quality Metrics
    overall_confidence DECIMAL(3,2) CHECK (overall_confidence BETWEEN 0
    missing_information TEXT[] DEFAULT '{}',
    warnings TEXT[] DEFAULT '{}',

    -- Status
    status VARCHAR(50) DEFAULT 'draft', -- 'draft', 'classification_pend

    -- Audit
    created_by UUID REFERENCES users(user_id),
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW(),
    finalized_at TIMESTAMP,

    -- Indexes
    CONSTRAINT valid_confidence CHECK (
```

```
        classification_confidence IS NULL OR (classification_confidence
        overall_confidence IS NULL OR (overall_confidence BETWEEN 0 AND
    )
);

CREATE INDEX idx_roi_models_client ON roi_models(client_id);
CREATE INDEX idx_roi_models_type ON roi_models(roi_model_type_id);
CREATE INDEX idx_roi_models_status ON roi_models(status);

-- ============================================================================
-- ROI Model History Table (Change Tracking)
-- ============================================================================
CREATE TABLE roi_model_history (
    history_id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    roi_model_id UUID NOT NULL REFERENCES roi_models(roi_model_id) ON DE

    -- Change Details
    changes_made JSONB NOT NULL DEFAULT '[]',
    user_prompt TEXT,
    refinement_type VARCHAR(50), -- 'direct_edit', 'prompt', 'scenario_a

    -- Audit
    changed_by UUID REFERENCES users(user_id),
    changed_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_roi_model_history_model ON roi_model_history(roi_model_
CREATE INDEX idx_roi_model_history_changed_at ON roi_model_history(chang

-- ============================================================================
-- Dashboard Templates Table (Updated to link to ROI Models)
-- ============================================================================
ALTER TABLE dashboard_templates
    ADD COLUMN roi_model_id UUID REFERENCES roi_models(roi_model_id) ON
    ADD COLUMN category VARCHAR(50), -- 'executive', 'financial', 'clini
    ADD COLUMN target_audience VARCHAR(50), -- 'Health Plan', 'Employer'
    ADD COLUMN display_order INTEGER DEFAULT 0,
```

```
    ADD COLUMN is_active BOOLEAN DEFAULT TRUE;

CREATE INDEX idx_dashboard_templates_roi_model ON dashboard_templates(ro
CREATE INDEX idx_dashboard_templates_category ON dashboard_templates(cat

-- ============================================================================
-- Prospect-ROI Model Association Table
-- ============================================================================
CREATE TABLE prospect_roi_models (
    prospect_id UUID REFERENCES prospects(prospect_id) ON DELETE CASCADE
    roi_model_id UUID REFERENCES roi_models(roi_model_id) ON DELETE CASC
    is_primary BOOLEAN DEFAULT FALSE,
    associated_at TIMESTAMP DEFAULT NOW(),
    associated_by UUID REFERENCES users(user_id),

    PRIMARY KEY (prospect_id, roi_model_id)
);

CREATE INDEX idx_prospect_roi_models_prospect ON prospect_roi_models(pro
CREATE INDEX idx_prospect_roi_models_roi_model ON prospect_roi_models(ro

-- Ensure only one primary ROI Model per prospect
CREATE UNIQUE INDEX idx_prospect_primary_roi_model
    ON prospect_roi_models(prospect_id)
    WHERE is_primary = TRUE;
```

Implementation Guide

Phase 1: Agent Development (Weeks 1-2)

Priority Order:

1. **ROI Story Classification Agent**
    - File: `agents/roi_story_classification_agent.py`
    - Instructions: `agents/templates/roi_story_classification_instructions.md`

- Models: `core/models/roi_model_models.py` (ROIStoryClassificationResult)
- Tests: `tests/test_roi_story_classification_agent.py`

2. **ROI Model Builder Agent**
   - File: `agents/roi_model_builder_agent.py`
   - Instructions: `agents/templates/roi_model_builder_instructions.md`
   - Type Templates: `agents/templates/roi_model_builders/type_01_unit_price_reduction.md` (all 13)
   - Tools: `tools/roi_model_calculator.py`
   - Models: `core/models/roi_model_models.py` (ROIModel, ROIModelVariable, ROIModelFormula)
   - Tests: `tests/test_roi_model_builder_agent.py`
3. **ROI Model Refinement Agent**
   - File: `agents/roi_model_refinement_agent.py`
   - Instructions: `agents/templates/roi_model_refinement_instructions.md`
   - Models: `core/models/roi_model_models.py` (ROIModelRefinementResult)
   - Tests: `tests/test_roi_model_refinement_agent.py`
4. **Update Dashboard Template Generator Agent**
   - Update: `agents/template_generator_agent.py` (accept ROI Model input)
   - Instructions: `agents/templates/dashboard_generation_from_roi_model_instructions.md`
   - Tests: `tests/test_dashboard_generator_with_roi_model.py`

Phase 2: API Development (Week 3)

Endpoints to Create:

1. ROI Model CRUD: `api/routes/roi_models.py`
2. Agent Execution: `api/routes/roi_model_agents.py`
3. Dashboard Generation: Update `api/routes/templates.py`
4. Prospect Association: Update `api/routes/prospects.py`

Phase 3: Database Migration (Week 4)

Migration Scripts:

1. ☐Create `roi_model_types` table
2. ☐Create `roi_models` table
3. ☐Create `roi_model_history` table
4. ☐Update `dashboard_templates` table
5. ☐Create `prospect_roi_models` table
6. ☐Migrate existing value propositions (if applicable)

Phase 4: Testing & Validation (Week 5)

Test Coverage:

- Unit tests for all 3 new agents
- Integration test for complete ROI Model creation flow
- API endpoint tests
- Database constraint tests
- Performance/load tests

Phase 5: Deployment (Week 6)

Deployment Steps:

1. ☐Deploy agent code to production
2. ☐Run database migrations
3. ☐Deploy updated API endpoints
4. ☐Update frontend to use new flow
5. ☐Monitor agent performance and error rates

Testing & Validation

Unit Test Examples

```
 # tests/test_roi_story_classification_agent.py
def test_pharmacy_optimization_classification():
    """Test classification of pharmacy-focused ROI Story."""
    agent = create_roi_story_classification_agent()

    roi_story = """
    Reduce specialty drug costs by 20% through dose optimization and adh
    Baseline pharmacy PMPM: $120. Specialty drugs represent 70% of pharm
    Target savings: $17 PMPM. Population: 50,000 members.
    """

    result = agent.run(roi_story)

    assert result.primary_roi_model_type == "Pharmacy Optimization"
    assert result.primary_type_id == 10
    assert result.confidence >= 0.8
    assert "specialty drug" in [k.lower() for k in result.key_indicators
    assert len(result.suggested_variables) >= 3

def test_low_confidence_classification():
    """Test classification with ambiguous ROI Story."""
    agent = create_roi_story_classification_agent()

    roi_story = "Reduce healthcare costs through operational improvement

    result = agent.run(roi_story)

    assert result.confidence < 0.7  # Low confidence expected
    assert len(result.secondary_types) >= 2  # Multiple types suggested
```

```python
 # tests/test_roi_model_builder_agent.py
def test_pharmacy_roi_model_building():
    """Test building complete ROI Model from Type 10."""
    agent = create_roi_model_builder_agent()

    input_data = {
        "roi_model_type_id": 10,
        "roi_story_content": "Baseline pharmacy PMPM: $120, Target: $96,
        "client_context": {"industry": "health_plan"}
    }

    result = agent.run(input_data)

    assert result.roi_model_type_id == 10
    assert "pharmacy_pmpm" in result.variables["baseline_costs"]
    assert result.variables["baseline_costs"]["pharmacy_pmpm"]["value"]
    assert result.formulas["annual_savings"]["result"] is not None
    assert result.overall_confidence >= 0.6

def test_formula_dependency_validation():
    """Test that formula dependencies are correctly validated."""
    agent = create_roi_model_builder_agent()

    # Create model with circular dependency (should fail)
    input_data_with_circular = {
        "roi_model_type_id": 10,
        "custom_formulas": {
            "savings": "cost - savings",  # Circular!
            "cost": "baseline + savings"
        }
    }

    with pytest.raises(ValueError, match="circular"):
        agent.run(input_data_with_circular)
```

Performance & Scalability

Performance Benchmarks

Operation	Target Duration	Token Usage	Success Rate
ROI Story Classification	15-30s	5K-10K	95%
ROI Model Building	30-60s	10K-20K	90%
ROI Model Refinement	10-20s	3K-8K	98%
Dashboard Generation	45-90s	15K-30K	92%
Total Flow (End-to-End)	**2-4 minutes**	**40K-70K**	**85%**

Scalability Considerations

Concurrent Users: - System should support 50+ concurrent ROI Model creation sessions - Agent execution should be parallelized (multiple jobs in background) - Database connection pooling required

Data Volume: - ROI Models: Up to 10,000 models per client - Dashboard Templates: Up to 100,000 templates (10 per model) - Prospects: Up to 1M prospects with associated data

Cost Optimization: - Cache classification results for similar ROI Stories - Reuse dashboard templates across similar prospects - Batch agent executions during off-peak hours

---

Document Status: □Comprehensive Technical Specification Complete **Next Action:** Review & Approve Architecture → Begin Phase 1 Implementation **Estimated Implementation Time:** 6 weeks (full system)

Feedback & Questions: Please review mermaid diagrams, agent flows, and API designs for accuracy and completeness.