

1 Stakeholder Analysis Matrix

There are various ways to use tables and typst programming to implement a stakeholder analysis matrix. Here are examples of how to do it.

- Fully manual Section 1.1
- Using a table helper Section 1.2
- From a CSV file Section 1.3
- From a JSON file Section 1.4
- From a YAML file Section 1.5

This is merely an example of the power of typst to create beautiful, professional documents in pdf and html.

1.1 Fully manual

Stakeholder	Pain (Top 5, validated)	Promise?	Proof (or Plan to get there)
Physician (plus nurse)	<ul style="list-style-type: none">• Time-consuming documentation• Alert fatigue• Workflow disruption• Limited clinical decision support• Integration challenges	Reduce documentation time by 50% while improving care quality	<ul style="list-style-type: none">• Pilot study with 10 physicians• Time-motion analysis• Satisfaction surveys• Clinical outcome tracking
Patient	<ul style="list-style-type: none">• Lack of transparency• Limited access to care• Complex navigation• Poor communication• High out-of-pocket costs	Improve patient engagement and reduce barriers to care access	<ul style="list-style-type: none">• Patient satisfaction scores• Access metrics• Engagement analytics• Cost analysis
Provider (Hospital)	<ul style="list-style-type: none">• Rising operational costs• Staff burnout• Quality metrics pressure• Regulatory compliance burden• Technology integration issues	Reduce costs by 20% while maintaining or improving quality metrics	<ul style="list-style-type: none">• Financial modeling• ROI analysis• Pilot program results• Benchmark comparisons
Payer	<ul style="list-style-type: none">• Unsustainable cost growth• Fraud and abuse• Quality measurement challenges• Administrative burden• Member satisfaction issues	Demonstrate cost savings and improved outcomes through value-based care	<ul style="list-style-type: none">• Claims data analysis• Cost-effectiveness studies• Quality metrics• Member retention data

Policy maker (Guidelines)	<ul style="list-style-type: none"> Limited evidence base Implementation gaps Health inequities Resource allocation challenges Regulatory complexity 	<p>Provide evidence-based solutions that scale and reduce health disparities</p>	<ul style="list-style-type: none"> Population health data Implementation studies Health equity metrics Cost-effectiveness analysis
--------------------------------------	--	--	--

1.2 Using a table helper

Stakeholder	Pain (Top 5, validated)	Promise?	Proof (or Plan to get there)
Physician (plus nurse)	<ul style="list-style-type: none"> Time-consuming documentation Alert fatigue Workflow disruption Limited clinical decision support Integration challenges 	<p>Reduce documentation time by 50% while improving care quality</p>	<ul style="list-style-type: none"> Pilot study with 10 physicians Time-motion analysis Satisfaction surveys Clinical outcome tracking
Patient	<ul style="list-style-type: none"> Lack of transparency Limited access to care Complex navigation Poor communication High out-of-pocket costs 	<p>Improve patient engagement and reduce barriers to care access</p>	<ul style="list-style-type: none"> Patient satisfaction scores Access metrics Engagement analytics Cost analysis
Provider (Hospital)	<ul style="list-style-type: none"> Rising operational costs Staff burnout Quality metrics pressure Regulatory compliance burden Technology integration issues 	<p>Reduce costs by 20% while maintaining or improving quality metrics</p>	<ul style="list-style-type: none"> Financial modeling ROI analysis Pilot program results Benchmark comparisons
Payer	<ul style="list-style-type: none"> Unsustainable cost growth Fraud and abuse Quality measurement challenges Administrative burden Member satisfaction issues 	<p>Demonstrate cost savings and improved outcomes through value-based care</p>	<ul style="list-style-type: none"> Claims data analysis Cost-effectiveness studies Quality metrics Member retention data
Policy maker (Guidelines)	<ul style="list-style-type: none"> Limited evidence base Implementation gaps Health inequities Resource allocation challenges Regulatory complexity 	<p>Provide evidence-based solutions that scale and reduce health disparities</p>	<ul style="list-style-type: none"> Population health data Implementation studies Health equity metrics Cost-effectiveness analysis

1.3 From a CSV file

Stakeholder	Pain (Top 5 validated)	Promise	Proof (or Plan to get there)
Physician (plus nurse)	<ul style="list-style-type: none"> • Time-consuming documentation • Alert fatigue • Workflow disruption • Limited clinical decision support • Integration challenges 	Reduce documentation time by 50% while improving care quality	Pilot study with 10 physiciansTime-motion analysisSatisfaction surveysClinical outcome tracking
Patient	<ul style="list-style-type: none"> • Lack of transparency • Limited access to care • Complex navigation • Poor communication • High out-of-pocket costs 	Improve patient engagement and reduce barriers to care access	Patient satisfaction scoresAccess metricsEngagement analyticsCost analysis
Provider (Hospital)	<ul style="list-style-type: none"> • Rising operational costs • Staff burnout • Quality metrics pressure • Regulatory compliance burden • Technology integration issues 	Reduce costs by 20% while maintaining or improving quality metrics	Financial modelingROI analysisPilot program resultsBenchmark comparisons
Payer	<ul style="list-style-type: none"> • Unsustainable cost growth • Fraud and abuse • Quality measurement challenges • Administrative burden • Member satisfaction issues 	Demonstrate cost savings and improved outcomes through value-based care	Claims data analysisCost-effectiveness studiesQuality metricsMember retention data
Policy maker (Guidelines)	<ul style="list-style-type: none"> • Limited evidence base • Implementation gaps • Health inequities • Resource allocation challenges • Regulatory complexity 	Provide evidence-based solutions that scale and reduce health disparities	Population health dataImplementation studiesHealth equity metricsCost-effectiveness analysis

1.4 From a JSON file

Stakeholder	Pain	Promise	Proof
Physician (plus nurse)	<ul style="list-style-type: none"> • Time-consuming documentation • Alert fatigue 	Reduce documentation time by 50% while improving care quality	<ul style="list-style-type: none"> • Pilot study with 10 physicians • Time-motion analysis

	<ul style="list-style-type: none"> Workflow disruption Limited clinical decision support Integration challenges 		<ul style="list-style-type: none"> Satisfaction surveys Clinical outcome tracking
Patient	<ul style="list-style-type: none"> Lack of transparency Limited access to care Complex navigation Poor communication High out-of-pocket costs 	Improve patient engagement and reduce barriers to care access	<ul style="list-style-type: none"> Patient satisfaction scores Access metrics Engagement analytics Cost analysis
Provider (Hospital)	<ul style="list-style-type: none"> Rising operational costs Staff burnout Quality metrics pressure Regulatory compliance burden Technology integration issues 	Reduce costs by 20% while maintaining or improving quality metrics	<ul style="list-style-type: none"> Financial modeling ROI analysis Pilot program results Benchmark comparisons
Payer	<ul style="list-style-type: none"> Unsustainable cost growth Fraud and abuse Quality measurement challenges Administrative burden Member satisfaction issues 	Demonstrate cost savings and improved outcomes through value-based care	<ul style="list-style-type: none"> Claims data analysis Cost-effectiveness studies Quality metrics Member retention data
Policy maker (Guidelines)	<ul style="list-style-type: none"> Limited evidence base Implementation gaps Health inequities Resource allocation challenges Regulatory complexity 	Provide evidence-based solutions that scale and reduce health disparities	<ul style="list-style-type: none"> Population health data Implementation studies Health equity metrics Cost-effectiveness analysis

1.5 From a YAML file

Stakeholder	Pain	Promise	Proof
Physician (plus nurse)	<ul style="list-style-type: none"> Time-consuming documentation Alert fatigue Workflow disruption Limited clinical decision support Integration challenges 	Reduce documentation time by 50% while improving care quality	<ul style="list-style-type: none"> Pilot study with 10 physicians Time-motion analysis Satisfaction surveys Clinical outcome tracking
Patient	<ul style="list-style-type: none"> Lack of transparency Limited access to care Complex navigation 	Improve patient engagement and reduce barriers to care access	<ul style="list-style-type: none"> Patient satisfaction scores Access metrics

	<ul style="list-style-type: none"> Poor communication High out-of-pocket costs 	<ul style="list-style-type: none"> Engagement analytics Cost analysis
Provider (Hospital)	<ul style="list-style-type: none"> Rising operational costs Staff burnout Quality metrics pressure Regulatory compliance burden Technology integration issues 	<p>Reduce costs by 20% while maintaining or improving quality metrics</p> <ul style="list-style-type: none"> Financial modeling ROI analysis Pilot program results Benchmark comparisons
Payer	<ul style="list-style-type: none"> Unsustainable cost growth Fraud and abuse Quality measurement challenges Administrative burden Member satisfaction issues 	<p>Demonstrate cost savings and improved outcomes through value-based care</p> <ul style="list-style-type: none"> Claims data analysis Cost-effectiveness studies Quality metrics Member retention data
Policy maker (Guidelines)	<ul style="list-style-type: none"> Limited evidence base Implementation gaps Health inequities Resource allocation challenges Regulatory complexity 	<p>Provide evidence-based solutions that scale and reduce health disparities</p> <ul style="list-style-type: none"> Population health data Implementation studies Health equity metrics Cost-effectiveness analysis

2 Technical Diagrams with Fletcher

The technical documentation package includes powerful diagram capabilities using Fletcher, a Typst library for creating flowcharts, architecture diagrams, and state machines. All diagrams support both PDF and HTML output with automatic dark mode adaptation.

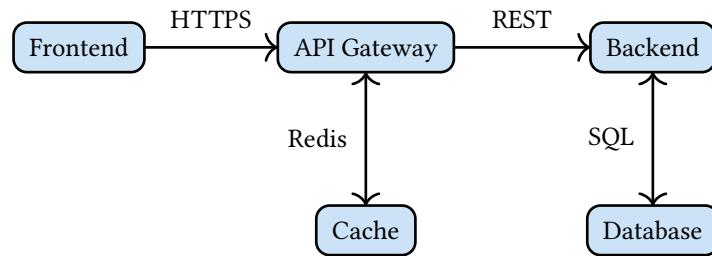
Dark Mode Support: When viewing in HTML, these diagrams automatically adapt to your system's color scheme using a centralized color palette defined in `lib/colors.json`. Try toggling dark mode to see the colors change seamlessly!

2.1 System Architecture Diagram

A system architecture diagram helps visualize how different components of a software system interact. This example shows a typical three-tier web application architecture with a frontend, API gateway, backend services, database, and caching layer.

Use cases:

- Documenting microservices architectures
- Planning infrastructure layouts
- Communicating system design to stakeholders
- Technical onboarding materials



The architecture diagram above illustrates:

- **Frontend:** User-facing web application
- **API Gateway:** Entry point for all client requests, handles routing and load balancing
- **Backend:** Application logic and business rules
- **Database:** Persistent data storage with SQL queries
- **Cache:** Redis layer for performance optimization

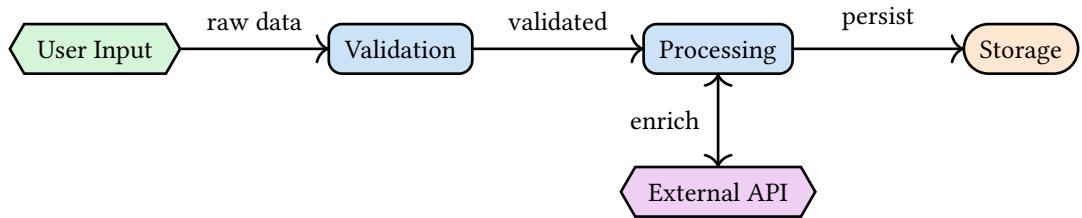
This pattern is commonly used in modern cloud-native applications, where each component can be scaled independently based on load requirements.

2.2 Data Flow Diagram

Data flow diagrams trace how information moves through a system, from input to output. They're essential for understanding data transformations, identifying bottlenecks, and planning data governance strategies.

Use cases:

- ETL (Extract, Transform, Load) pipeline documentation
- Data privacy impact assessments
- API integration specifications
- Quality assurance planning



The data flow shows a typical processing pipeline:

1. **User Input:** Raw data enters the system
2. **Validation:** Input is checked for correctness and security
3. **Processing:** Core business logic is applied
4. **External API:** Data is enriched with third-party information
5. **Storage:** Final results are persisted to the database

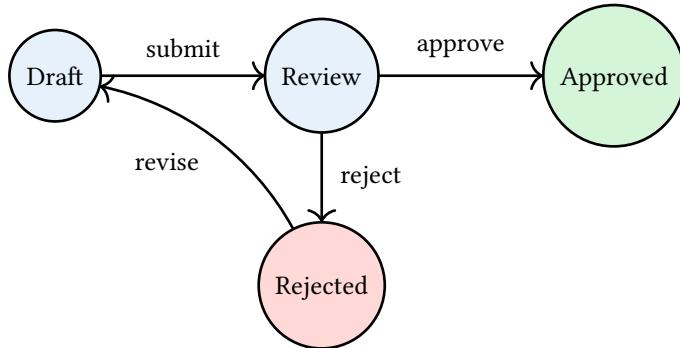
This pattern ensures data quality and enables traceability throughout the system. Each transformation step can be monitored, logged, and audited for compliance requirements.

2.3 State Machine Diagram

State machines model how systems transition between different states in response to events. They're invaluable for workflow design, business process modeling, and ensuring state consistency in distributed systems.

Use cases:

- Order fulfillment workflows
- Document approval processes
- User authentication flows
- IoT device lifecycle management



The workflow state machine represents a document review process:

- **Draft:** Initial creation phase where authors can freely edit
- **Review:** Document is submitted for peer review and approval
- **Approved:** Content has passed review and is published
- **Rejected:** Review identified issues requiring revision

Transitions between states are triggered by user actions (submit, approve, reject) or system events (timeout, validation failure). The "revise" transition creates a feedback loop, allowing iterative improvement until approval is achieved.

2.4 Benefits of Diagram-Driven Documentation

Using diagrams in technical documentation provides several advantages:

Clarity: Visual representations often communicate complex concepts more effectively than text alone. A single diagram can replace pages of description.

Maintenance: Diagrams created with code (like these Fletcher examples) can be version-controlled, diffed, and automatically regenerated when system designs change.

Accessibility: Modern diagram tools produce semantic markup that works with screen readers and other assistive technologies.

Consistency: Using a standardized diagramming library ensures visual consistency across all documentation, making it easier for readers to understand different sections.

Collaboration: Text-based diagram definitions can be reviewed in pull requests just like code, enabling better team collaboration.

2.5 Customizing Diagrams

These diagrams are defined in simple Typst code and can be customized with different layouts, labels, and spacing to match your specific use case.

Color System: All diagram colors come from `lib/colors.json`, which defines light/dark mode pairs.

Using arbitrary colors will break dark mode! To customize colors:

- Edit `lib/colors.json` to add/modify color pairs
- Run `make colors` to regenerate CSS and Typst files
- Use the predefined colors: `node-bg-blue`, `node-bg-green`, `node-bg-orange`, `node-bg-purple`, `node-bg-red`, `node-bg-neutral`

See `docs/DARK_MODE_COLOR_STANDARDS.md` for details on the color system and `technical-documentation/README.md` for diagram examples.