

# Bridgelet

MVP Specification (v0.1)

**Version:** 0.1

**Target Release:** Q1 2026

**Status:** Early Development

**Last Updated:** January 2025

## 1. Executive Summary

Bridgelet MVP focuses on delivering core functionality for creating single-use ephemeral Stellar accounts that enable cryptocurrency payments to recipients without existing wallets. The MVP prioritizes simplicity, security, and a working end-to-end flow over advanced features.

### 1.1 Success Criteria

- Organizations can create ephemeral accounts programmatically
- Recipients can claim funds with minimal friction
- Funds auto-sweep to recipient's permanent wallet
- Unclaimed funds return to organization after expiration
- System handles 100+ accounts per organization

## 2. Core Features (In Scope)

**Feature 1: Ephemeral Account Creation**

**Description:** Organizations can create temporary Stellar accounts via SDK

**Requirements:**

- REST API endpoint for account creation
- Support for XLM and USDC assets
- Configurable expiration time (1-90 days)
- Unique claim code generation
- Email/phone number for recipient

**Acceptance Criteria:**

- Account created on Stellar testnet/mainnet
- Claim URL generated and returned
- Account metadata stored in database
- Expiration timestamp set correctly

## Feature 2: Single-Use Account Restriction

**Description:** Smart contract enforces single inbound payment and single outbound sweep

**Requirements:**

- Soroban contract restricts account operations
- Only one payment can be received
- Only one sweep transaction allowed
- Destination wallet locked during creation

**Acceptance Criteria:**

- Contract deployed and tested on testnet
- Multiple payment attempts rejected
- Sweep only to authorized destination
- Account closed after successful sweep

## Feature 3: Claim Flow

**Description:** Recipients can claim funds using a simple web interface

**Requirements:**

- Claim page accessible via unique URL
- Email/SMS verification
- Wallet connection (Freighter, Albedo)
- First-time wallet creation guidance
- Automatic sweep upon successful claim

**Acceptance Criteria:**

- Claim link works in all modern browsers
- Verification code validates correctly
- Wallets connect without errors
- Funds transferred to recipient wallet
- Confirmation displayed to user

## **Feature 4: Expiration & Recovery**

**Description:** Unclaimed funds automatically return to organization after expiration

**Requirements:**

- Automated expiration checker (cron job)
- Recovery transaction to organization wallet
- Notification to organization
- Account status updated

**Acceptance Criteria:**

- Expired accounts detected within 1 hour
- Recovery transaction executed successfully
- Organization receives funds minus network fees
- Account marked as recovered

## Feature 5: SDK & API

**Description:** TypeScript SDK for Node.js with REST API backend

### **Requirements:**

- `createEphemeralAccount()` method
- `getAccountStatus()` method
- `listAccounts()` method
- `recoverExpiredAccount()` method
- API key authentication
- Rate limiting (100 req/min)

### **Acceptance Criteria:**

- SDK published to npm
- API documentation complete
- Example code provided
- Integration tests passing

### 3. Excluded Features (Out of Scope for MVP)

#### ✖ Multi-Asset Support

MVP supports only XLM and USDC. Other assets postponed to v0.2

#### ✖ Batch Operations

Bulk account creation will be manual iteration in MVP. Batch API in v0.2

#### ✖ Advanced Authentication

OAuth2, SSO, and MFA postponed. MVP uses API keys only

#### ✖ Webhooks

Event notifications via webhooks postponed to v0.2

#### ✖ Mobile Apps

Native iOS/Android apps not included. Web-only for MVP

#### ✖ Advanced Analytics

Dashboard with charts and reporting postponed. Basic status checks only

## 4. Technical Requirements

### 4.1 Smart Contracts (bridgelet-core)

Component	Technology	Status
Account Restriction Contract	Soroban (Rust)	In Progress
Single-Use Enforcement	Smart Contract Logic	In Progress
Auto-Sweep Logic	Smart Contract Logic	Pending
Expiration Handling	Smart Contract Logic	Pending

### 4.2 Backend SDK (bridgelet-sdk)

Component	Technology	Status
REST API	NestJS	In Progress
Database	PostgreSQL	In Progress
Authentication	JWT	Pending
Stellar Integration	Stellar SDK	In Progress
Expiration Worker	Cron Jobs	Pending

### 4.3 Frontend (bridgelet-ui)

Component	Technology	Status
Claim Page	Next.js	Pending
Wallet Connection	Stellar Wallets Kit	Pending
Email Verification	OTP System	Pending
Status Tracking	React Components	Pending

## 5. Development Timeline

### ⌚ Milestone 1: Smart Contract Foundation (Weeks 1-3)

- Design contract architecture
- Implement single-use restrictions
- Deploy to testnet
- Write comprehensive tests

**Deliverable:** Working smart contract on Stellar testnet

### ⌚ Milestone 2: SDK & API Development (Weeks 4-6)

- Set up NestJS backend
- Implement core API endpoints
- Create TypeScript SDK
- Set up PostgreSQL database
- Write integration tests

**Deliverable:** Functional API with SDK

### ⌚ Milestone 3: Claim UI (Weeks 7-8)

- Build claim page
- Integrate wallet connections
- Implement verification flow
- Add error handling

**Deliverable:** Working claim interface

### ⌚ Milestone 4: Expiration & Recovery (Weeks 9-10)

- Build expiration detector
- Implement recovery logic
- Set up automated jobs
- Test edge cases

**Deliverable:** Complete lifecycle management

### 🎯 Milestone 5: Testing & Documentation (Weeks 11-12)

- End-to-end testing
- Security review
- Write documentation
- Create example integrations
- Bug fixes

**Deliverable:** Production-ready MVP

## 6. Success Metrics

### 6.1 Technical Metrics

Metric	Target	Measurement
Account Creation Time	< 5 seconds	API response time
Claim Success Rate	> 95%	Claims / Attempts
API Uptime	> 99.5%	Monitoring service
Transaction Cost	< \$0.10	Stellar network fees

### 6.2 User Experience Metrics

Metric	Target	Measurement
Claim Completion Rate	> 70%	Completed / Started
Time to First Claim	< 5 minutes	User analytics
Support Tickets	< 5% of users	Support system

## 7. Risk Assessment

---

### 7.1 Technical Risks

Risk	Probability	Impact	Mitigation
Smart contract bugs	Medium	High	Comprehensive testing, audit
Stellar network issues	Low	High	Fallback strategies, monitoring
Scaling challenges	Medium	Medium	Load testing, incremental rollout

### 7.2 User Adoption Risks

- **Risk:** Recipients don't understand crypto wallets  
**Mitigation:** Clear onboarding, video tutorials, support
- **Risk:** Low claim rates  
**Mitigation:** Multi-channel notifications, longer expiration windows
- **Risk:** Security concerns from organizations  
**Mitigation:** Security audit, insurance options, clear documentation

## 8. Post-MVP Roadmap

---

## **Version 0.2 (Q2 2026)**

- Batch account creation API
- Webhook notifications
- Analytics dashboard
- Additional asset support
- Mobile-optimized UI

## **Version 1.0 (Q3 2026)**

- Mainnet production launch
- Security audit completion
- Native mobile apps
- Advanced authentication (OAuth2)
- White-label solutions

---

### **Bridgelet MVP Specification v0.1**

This specification is a living document and will be updated as development progresses.

Last updated: January 2025