Bridgelet

Use Cases & Real-World Examples

> **Document Version:** 0.1 (MVP)
> **Last Updated:** January 2025
> **Purpose:** Demonstrate practical applications of Bridgelet

1. Overview

Bridgelet enables organizations to send cryptocurrency payments to recipients without requiring them to have existing wallets or understand blockchain technology. This opens up numerous use cases across various industries.

2. Payroll & Employee Compensation

💼 Use Case: Remote Worker Payroll in Emerging Markets

> **Challenge:**
> A global company employs remote workers in countries with limited banking infrastructure. Traditional international wire transfers are expensive ($25-50 per transaction), slow (3-5 days), and many workers lack bank accounts.

> **Bridgelet Solution:**

1. Company integrates Bridgelet SDK with their payroll system
2. On payday, system creates ephemeral accounts for all employees
3. Employees receive SMS/email with claim links
4. First-time users create wallet while claiming (one-time setup)
5. Funds instantly transferred to their wallets
6. Employees can cash out via local exchanges or use for payments

Results:

✅ Cost: $0.01-0.10 per transaction (vs $25-50)
✅ Speed: Instant (vs 3-5 days)
✅ Accessibility: No bank account required
✅ Onboarding: 80% of employees set up wallets within first pay cycle

Implementation Example

```
// Monthly payroll integration
async function processPayroll(employees) {
  const paymentPromises = employees.map(async (employee)
    const account = await bridgelet.createEphemeralAccoun
      amount: employee.salary.toString(),
      asset: 'USDC',
      recipientEmail: employee.email,
      recipientPhone: employee.phone,
      expiresIn: '30d',
      metadata: {
        employeeId: employee.id,
        payPeriod: '2026-01',
        department: employee.department
      }
    });

    // Send claim link via SMS and email
    await Promise.all([
```

```
      sendSMS(employee.phone, `Your salary is ready! Clai
      sendEmail(employee.email, 'payroll-notification', {
    ]);

    return account;
  });

  return await Promise.all(paymentPromises);
}
```

3. Humanitarian Aid Distribution

🌍 Use Case: Disaster Relief Cash Transfers

Challenge:
NGO needs to distribute emergency cash assistance to displaced families in a disaster zone. Recipients have limited access to banking, internet connectivity is poor, and traditional aid distribution is slow and fraud-prone.

Bridgelet Solution:

1. NGO collects phone numbers through registration centers
2. Creates ephemeral accounts with aid amounts
3. Sends claim links via SMS (works on basic phones)
4. Recipients visit local cryptocurrency kiosks or use mobile wallets
5. Claim funds using SMS verification code
6. Convert to local currency or use directly for purchases

Results:
✅ Distribution Speed: 1000+ families per hour
✅ Fraud Reduction: Blockchain transparency eliminates double-dipping
✅ Cost Efficiency: 95% of funds reach recipients (vs 70% traditional)
✅ Audit Trail: Complete transparency for donors

Implementation Example

```
 // Batch aid distribution
async function distributeCashAid(recipients, amountPerRec
  const accounts = await bridgelet.createBulkAccounts(
    recipients.map(r => ({
      amount: amountPerRecipient,
      asset: 'USDC',
      recipientPhone: r.phone,
      recipientEmail: r.email,
      expiresIn: '60d', // Longer window for disaster sce
      metadata: {
        beneficiaryId: r.id,
        program: 'emergency-relief-2026',
        location: r.location
      }
    }))
  );

  // Send SMS notifications in batches
  const smsQueue = accounts.map(acc => ({
    to: acc.recipientPhone,
    message: `Emergency aid of $${amountPerRecipient} is
  }));

  await sendBulkSMS(smsQueue);

  // Generate distribution report
  await generateReport({
    totalRecipients: recipients.length,
    totalAmount: amountPerRecipient * recipients.length,
    accounts: accounts
  });
}
```

4. Creator Economy & Micropayments

🎨 Use Case: Content Creator Payouts

Challenge:
A video platform wants to pay creators from ad revenue shares. Creators are globally distributed, many lack traditional payment infrastructure, and micropayments ($1-50) are not economical with traditional payment rails.

Bridgelet Solution:

1. Platform calculates monthly earnings for each creator
2. Creates ephemeral accounts for payouts
3. Creators claim funds when they reach minimum threshold
4. First-time claimants set up wallet during claim process
5. Repeat claimants receive instant transfers

Results:
✅ Minimum Payout: $1 (vs $100 with traditional systems)
✅ Payout Frequency: Weekly (vs monthly/quarterly)
✅ Creator Satisfaction: 90% prefer crypto payouts
✅ Platform Costs: 80% reduction in payment processing fees

Implementation Example

```
// Creator payout system
async function processCreatorPayouts() {
```

```
  const creators = await getCreatorsWithEarnings();

  for (const creator of creators) {
    if (creator.earnings >= creator.minimumPayout) {
      const account = await bridgelet.createEphemeralAcco
        amount: creator.earnings.toFixed(2),
        asset: 'USDC',
        recipientEmail: creator.email,
        expiresIn: '90d',
        metadata: {
          creatorId: creator.id,
          period: getCurrentPeriod(),
          views: creator.totalViews,
          adRevenue: creator.earnings
        }
      });

      // Notify creator
      await sendEmail(creator.email, 'payout-ready', {
        amount: creator.earnings,
        claimUrl: account.claimUrl,
        period: getCurrentPeriod()
      });

      // Mark as paid in database
      await markCreatorAsPaid(creator.id, account.id);
    }
  }
}
```

5. Airdrops & Token Distribution

🪂 Use Case: NFT Project Community Airdrop

> **Challenge:**
> NFT project wants to distribute tokens to 10,000 community members. Many are new to crypto, and requiring them to set up wallets before airdrop creates friction and reduces participation.

Bridgelet Solution:

1. Project creates ephemeral accounts for all eligible recipients
2. Sends claim links via social media/email
3. Recipients click link and are guided through wallet setup
4. Tokens automatically transferred to new or existing wallet
5. Unclaimed tokens returned to project after expiration

Results:

✅ Claim Rate: 75% (vs 30% with traditional airdrops)
✅ New User Onboarding: 60% created first wallet through claim
✅ Time Saved: 2 weeks vs 3 months for manual distributions
✅ Fraud Prevention: One claim per verified account

6. Marketplace & Gig Economy Payouts

🛒 Use Case: Freelance Marketplace Payments

Challenge:
Freelance platform connects clients with service providers globally. Providers face high fees (10-20%), slow payouts (7-14 days), and currency conversion issues.

Bridgelet Solution:

1. Client completes payment on platform
2. Platform creates ephemeral account for service provider
3. Provider notified of payment
4. Funds claimed instantly to their wallet
5. Provider can hold crypto or cash out locally

Results:

✅ Provider Fees: 2% (vs 10-20%)
✅ Payout Speed: Instant (vs 7-14 days)
✅ Provider Retention: 40% improvement
✅ Cross-border Friction: Eliminated

Implementation Example

```
// Escrow release with Bridgelet
async function releaseEscrowPayment(jobId) {
  const job = await getJobDetails(jobId);

  // Create ephemeral account for service provider
  const account = await bridgelet.createEphemeralAccount(
```

```
    amount: job.amount.toString(),
    asset: 'USDC',
    recipientEmail: job.providerEmail,
    expiresIn: '30d',
    metadata: {
      jobId: job.id,
      clientId: job.clientId,
      providerId: job.providerId,
      completionDate: new Date().toISOString()
    }
  });

  // Update job status
  await updateJobStatus(jobId, 'payment_released');

  // Notify provider
  await sendEmail(job.providerEmail, 'payment-released',
    amount: job.amount,
    claimUrl: account.claimUrl,
    jobTitle: job.title
  });

  return account;
}
```

7. Implementation Comparison

Use Case	Traditional Method	With Bridgelet	Key Benefit
Payroll	Wire transfers, $25-50/tx, 3-5 days	Crypto, $0.01/tx, instant	99% cost reduction
Aid Distribution	Cash handouts, 30% overhead	Digital claims, 5% overhead	Transparency & efficiency

Use Case	Traditional Method	With Bridgelet	Key Benefit
Creator Payouts	$100 min, monthly	$1 min, weekly	Accessibility
Airdrops	30% claim rate	75% claim rate	User onboarding
Marketplace	7-14 days, 10-20% fees	Instant, 2% fees	Provider retention

8. Anti-Patterns (What NOT to Do)

8.1 Don't Use Bridgelet For:

- ❌ Long-term storage (ephemeral accounts expire)
- ❌ Multiple recurring payments to same recipient (use direct wallet transfer)
- ❌ Situations where recipient already has wallet (unnecessary friction)
- ❌ High-frequency microtransactions (use payment channels instead)

8.2 Common Mistakes

- ❌ Setting expiration too short (recipients need time to claim)
- ❌ Not providing fallback communication methods
- ❌ Forgetting to handle expired account recovery
- ❌ Not educating recipients about wallet security

9. Getting Started with Your Use Case

Step 1: Identify Your Requirements

- How many recipients per batch?
- Payment frequency?

- Average payment amount?
- Recipient technical literacy?
- Preferred communication channels?

Step 2: Plan Your Integration

- Map Bridgelet into your existing payment flow
- Design recipient communication strategy
- Set appropriate expiration windows
- Plan for claim support and education

Step 3: Pilot Program

- Start with small user group (10-100 recipients)
- Gather feedback on claim experience
- Measure success metrics
- Iterate before full rollout
