

Deployment Guide - QuantPayChain MVP

⚠ AUTOMATIC TRANSLATION — REVIEW REQUIRED

This document has been automatically translated from Spanish. Technical review is recommended for accuracy.

Table of Contents

1. Prerequisites
2. Contract Deployment
3. Frontend Deployment
4. Environment Variables Configuration
5. Deployment Verification
6. Troubleshooting

Prerequisites

Required Software

- **Node.js:** v18.x or higher
- **npm:** v9.x or higher
- **Foundry:** Latest stable version
- **Git:** For repository cloning
- **MetaMask:** Configured browser extension

Accounts and Services

- **Ethereum Wallet:** With funds on Sepolia testnet
- **Alchemy/Infura:** API key for RPC provider
- **Etherscan:** API key for contract verification (optional)
- **Vercel:** Account for frontend deployment (optional)

Getting Testnet ETH

```
# Recommended faucets for Sepolia:
# - https://sepoliafaucet.com/
# - https://www.alchemy.com/faucets/ethereum-sepolia
# - https://faucet.quicknode.com/ethereum/sepolia
```

Contract Deployment

1. Clone Repository

```
git clone https://github.com/francoMengarelli/quantpaychain-mvp.git
cd quantpaychain-mvp/contracts
```

2. Install Dependencies

```
# Install Foundry dependencies
forge install

# Verify installation
forge --version
```

3. Configure Environment Variables

Create `.env` file in `contracts/` directory:

```
# RPC URL for Sepolia
SEPOLIA_RPC_URL=https://eth-sepolia.g.alchemy.com/v2/YOUR_API_KEY

# Deployment wallet private key (NEVER share!)
PRIVATE_KEY=0xYOUR_PRIVATE_KEY_HERE

# Etherscan API key for verification
ETHERSCAN_API_KEY=YOUR_ETHERSCAN_API_KEY

# Deployer address (your wallet)
DEPLOYER_ADDRESS=0xYOUR_WALLET_ADDRESS
```

4. Compile Contracts

```
# Compile all contracts
forge build

# Verify no errors
forge test
```

5. Execute Deployment

```
# Deploy to Sepolia testnet
forge script script/Deploy.s.sol:DeployScript \
  --rpc-url $SEPOLIA_RPC_URL \
  --private-key $PRIVATE_KEY \
  --broadcast \
  --verify \
  --etherscan-api-key $ETHERSCAN_API_KEY

# Script will display deployed contract addresses
```

6. Save Contract Addresses

Deployment script will generate a file with addresses:

```
{
  "PaymentProcessor": "0x...",
  "TokenManager": "0x...",
  "DisputeResolver": "0x...",
  "Governance": "0x..."
}
```

⚠️ IMPORTANT: Save these addresses for frontend configuration.

Frontend Deployment

1. Navigate to Frontend Directory

```
cd ../frontend
```

2. Install Dependencies

```
npm install
```

3. Configure Environment Variables

Create `.env.local` file:

```
# Contract addresses (from previous step)
NEXT_PUBLIC_PAYMENT_PROCESSOR_ADDRESS=0x...
NEXT_PUBLIC_TOKEN_MANAGER_ADDRESS=0x...
NEXT_PUBLIC_DISPUTE_RESOLVER_ADDRESS=0x...
NEXT_PUBLIC_GOVERNANCE_ADDRESS=0x...

# Network configuration
NEXT_PUBLIC_CHAIN_ID=11155111
NEXT_PUBLIC_RPC_URL=https://eth-sepolia.g.alchemy.com/v2/YOUR_API_KEY

# Application configuration
NEXT_PUBLIC_APP_NAME=QuantPayChain
NEXT_PUBLIC_APP_URL=https://your-domain.vercel.app
```

4. Local Development

```
# Start development server
npm run dev

# Application will be available at http://localhost:3000
```

5. Production Build

```
# Create optimized build
npm run build

# Test build locally
npm start
```

6. Deploy to Vercel

Option A: From CLI

```
# Install Vercel CLI
npm i -g vercel

# Login to Vercel
vercel login

# Deploy
vercel --prod
```

Option B: From GitHub

1. Connect repository at vercel.com (<https://vercel.com>)
2. Configure environment variables in dashboard
3. Vercel will automatically deploy on each push

7. Configure Variables in Vercel Dashboard

1. Go to Project Settings → Environment Variables
2. Add all variables from `.env.local`
3. Select environments: Production, Preview, Development
4. Save and redeploy if necessary

Environment Variables Configuration

Critical Variables

Contracts

```
# Never commit these variables
PRIVATE_KEY=          # Deployer private key
SEPOLIA_RPC_URL=      # RPC provider URL
ETHERSCAN_API_KEY=    # For contract verification
```

Frontend

```
# Public variables (NEXT_PUBLIC_ prefix)
NEXT_PUBLIC_PAYMENT_PROCESSOR_ADDRESS= # Contract address
NEXT_PUBLIC_TOKEN_MANAGER_ADDRESS=    # Contract address
NEXT_PUBLIC_DISPUTE_RESOLVER_ADDRESS= # Contract address
NEXT_PUBLIC_GOVERNANCE_ADDRESS=       # Contract address
NEXT_PUBLIC_CHAIN_ID=                # Network ID (11155111 for Sepolia)
NEXT_PUBLIC_RPC_URL=                 # Public RPC URL
```

Variable Security

- Use `.env.example` as template
- Add `.env*` to `.gitignore`
- Rotate keys regularly
- Never commit `.env` files
- Never share private keys

Deployment Verification

1. Verify Contracts on Etherscan

```
# Visit Sepolia Etherscan
https://sepolia.etherscan.io/address/YOUR_CONTRACT_ADDRESS

# Verify that:
# - Contract is verified (✓ green icon)
# - Source code is visible
# - Deployment transactions are successful
```

2. Test Frontend

```
# Functionality checklist:
# ✓ MetaMask connection works
# ✓ Language switching (ES/EN) works
# ✓ Page navigation works
# ✓ Contract interaction works (in demo mode)
# ✓ Responsive design on mobile/tablet/desktop
```

3. Run Tests

```
# Contract tests
cd contracts
forge test -vvv

# Frontend tests (if exist)
cd ../frontend
npm test
```

4. Verify CI/CD

```
# Review GitHub Actions
# - Contract build: ✓
# - Contract tests: ✓
# - Frontend build: ✓
# - Automatic deployment: ✓
```

Troubleshooting

Common Issues

Error: “Insufficient funds”

```
# Solution: Get more testnet ETH
# Visit Sepolia faucets and request funds
```

Error: “Nonce too low”

```
# Solution: Reset nonce in MetaMask
# Settings → Advanced → Reset Account
```

Error: “Contract verification failed”

```
# Solution: Verify manually
forge verify-contract \
--chain-id 11155111 \
--compiler-version v0.8.20 \
YOUR_CONTRACT_ADDRESS \
src/YourContract.sol:YourContract \
--etherscan-api-key $ETHERSCAN_API_KEY
```

Error: “RPC rate limit exceeded”

```
# Solution: Use RPC provider with higher limit
# - Alchemy: 300 req/s on free plan
# - Infura: 100,000 req/day on free plan
```

Error: “Module not found” in Frontend

```
# Solution: Clear cache and reinstall
rm -rf node_modules .next
npm install
npm run build
```

Logs and Debugging

View contract deployment logs

```
# Foundry saves logs in:
broadcast/Deploy.s.sol/11155111/run-latest.json
```

View Vercel logs

```
# From CLI
vercel logs

# From dashboard
# Project → Deployments → [Select deployment] → Logs
```

Support Contact

If problems persist:

1. Review [GitHub Issues](https://github.com/francoMengarelli/quantpaychain-mvp/issues) (<https://github.com/francoMengarelli/quantpaychain-mvp/issues>)
2. Create new issue with:
 - Problem description
 - Relevant logs
 - Steps to reproduce
 - Environment (OS, software versions)

Final Deployment Checklist

- [] Contracts compiled without errors
- [] Contract tests passing (59/59)

- [] Contracts deployed on Sepolia
 - [] Contracts verified on Etherscan
 - [] Environment variables configured
 - [] Frontend deployed on Vercel
 - [] MetaMask connection functional
 - [] Internationalization working
 - [] Responsive design verified
 - [] CI/CD configured and working
 - [] Documentation updated
 - [] README with clear instructions
-

Last updated: October 2025

Version: 1.0.0