XyloNet Complete Implementation Plan & Enhancement Roadmap

> ***Full development roadmap covering all completed phases (1-5) and proposed enhancements (6-9) for Arc Builders Fund alignment***

Table of Contents

Part A: Completed Implementation

Part B: Proposed Enhancements

Part C: Technical Reference

Implementation Status Overview

Phase	Description	Status	Completion
Phase 1	Foundation (Contracts + Deployment)	✅ Complete	100%
Phase 2	Bridge Integration (CCTP V2)	✅ Complete	100%
Phase 3	Yield Features (ERC-4626 Vault)	✅ Complete	100%
Phase 4	UX/UI Polish & Mobile	✅ Complete	100%
Phase 5	Documentation & Launch Prep	✅ Complete	100%
Phase 6	Core Arc Integrations	🔄 Proposed	0%
Phase 7	Capital Formation Features	🔄 Proposed	0%

Phase 8	Advanced Optimizations	🔄 Proposed	0%
Phase 9	Governance & Token	🔄 Proposed	0%

PART A: COMPLETED IMPLEMENTATION

Phase 1: Foundation ✅ COMPLETE

1.1 Project Setup

- ☑ Initialize Foundry/Hardhat project structure
- ☑ Initialize Next.js 16 frontend with TypeScript
- ☑ Configure Arc Testnet connection (Chain ID: 5042002)
- ☑ Setup environment variables and secrets
- ☑ Configure wagmi v2 and RainbowKit

1.2 Core Smart Contracts

- ☑ **XyloFactory.sol** - Pool deployment factory
  - createPool(), getPool(), setSwapFee(), setFeeRecipient()
- ☑ **XyloStablePool.sol** - StableSwap AMM implementation
  - Curve invariant with A=200
  - swap(), addLiquidity(), removeLiquidity()
- ☑ **XyloRouter.sol** - Trade routing and execution
  - swap(), addLiquidity(), removeLiquidity(), getAmountOut()
- ☑ **XyloERC20.sol** - LP token implementation
  - ERC20 standard with permit support

1.3 Interface Contracts

- ☑ IERC20.sol - ERC20 interface with Permit
- ☑ IXyloFactory.sol - Factory interface
- ☑ IXyloPool.sol - Pool interface
- ☑ IXyloRouter.sol - Router interface

1.4 Deployment

- ☑ Deploy XyloFactory to Arc Testnet
- ☑ Deploy XyloRouter to Arc Testnet
- ☑ Verify all contracts on Blockscout
- ☑ Create USDC-EURC Pool
- ☑ Create USDC-USYC Pool
- ☑ Add initial liquidity to pools

Phase 2: Bridge Integration ✅ COMPLETE

2.1 CCTP Bridge Contract

- ☑ **XyloBridge.sol** - CCTP V2 wrapper contract
  - bridgeOut() - Initiate cross-chain transfer
  - bridgeToChain() - Bridge by chain name
  - Support for depositForBurn with maxFee and hookData
- ☑ Deploy XyloBridge to Arc Testnet
- ☑ Verify on Blockscout

2.2 Bridge Frontend

- ☑ Chain selector component with official logos
- ☑ Bridge transaction flow with 4-step progress
- ☑ Real-time status tracking via events
- ☑ Circle Bridge Kit integration for auto-delivery
- ☑ Fast transfer mode support (~30 seconds)
- ☑ Error handling and retry logic

2.3 Supported Chains

- ☑ Ethereum Sepolia (Domain 0)
- ☑ Arbitrum Sepolia (Domain 3)
- ☑ Base Sepolia (Domain 6)
- ☑ Optimism Sepolia (Domain 2)
- ☑ Polygon Amoy (Domain 7)
- ☑ Avalanche Fuji (Domain 1)
- ☑ Arc Testnet (Domain 26)

Phase 3: Yield Features ✅ COMPLETE

3.1 Vault Contract

- ☑ **XyloVault.sol** - ERC-4626 compliant vault
  - deposit(), withdraw(), redeem()
  - previewDeposit(), previewWithdraw()
  - totalAssets(), convertToShares(), convertToAssets()
- ☑ Fee structure (0% deposit, 0.1% withdraw, 10% performance)
- ☑ Deploy to Arc Testnet
- ☑ Verify on Blockscout

3.2 Vault Frontend

- ☑ Vault dashboard with stats
- ☑ Deposit/withdraw modals
- ☑ APY calculation (dynamic based on TVL)
- ☑ Real-time balance updates
- ☑ Position tracking (shares → USDC value)

Phase 4: UX/UI Polish & Mobile ✅ COMPLETE

4.1 Design System

- ☑ Orbiter Finance-inspired dark theme
- ☑ CSS custom properties for theming
- ☑ Gradient accents and glass effects
- ☑ Consistent component styling
- ☑ Card hover animations

4.2 Enhanced Components

- ☑ **SwapWidget** - MAX button, USD values, price impact
- ☑ **BridgeWidget** - Chain logos, progress steps, status
- ☑ **Pools page** - TVL sparklines, analytics cards
- ☑ **Vault page** - APY sparkline, position tracking
- ☑ **History page** - Filter tabs, status badges

4.3 UI Components

- ☑ **Toast** - Transaction notifications
- ☑ **TokenLogo** - Official USDC, EURC, USYC logos
- ☑ **ChainLogo** - Official chain logos (Ethereum, Arbitrum, etc.)
- ☑ **Skeleton** - Loading states
- ☑ **Tooltip** - Info tooltips for DeFi terms
- ☑ **Confetti** - Success celebrations
- ☑ **EmptyState** - No data states

4.4 Mobile Responsiveness

- ☑ **Header.tsx** - Compact mobile navigation, hamburger menu
- ☑ **SwapWidget.tsx** - Responsive font sizes, touch targets
- ☑ **BridgeWidget.tsx** - Stacked layout, bottom sheet modal
- ☑ **Pools page** - Card view on mobile, grid stats
- ☑ **Vault page** - Responsive stats, stacked buttons
- ☑ **History page** - Card list on mobile, compact filters
- ☑ **Footer.tsx** - Reordered for mobile
- ☑ **globals.css** - Mobile-specific CSS (200+ lines)
    - Touch targets 44-48px minimum
    - Safe area insets for notched devices
    - Active state feedback (scale animations)
    - Horizontal scroll for filters
    - Bottom sheet modals

4.5 Revolutionary UI/UX Enhancements

- ☑ **AnimatedBackground.tsx** - Canvas particle system with aurora effect
    - 120+ animated particles with connection lines
    - Aurora borealis gradient effect
    - Mouse interaction (particles attracted to cursor)

    - GPU-accelerated with requestAnimationFrame
- ☑ **TiltCard.tsx** - 3D perspective cards
    - Customizable tilt amount (0-20°)
    - Dynamic glare effect following mouse
    - Holographic shine animation
    - Floating effect on hover
- ☑ **AnimatedElements.tsx** - Interactive elements
    - AnimatedNumber: Counting animations with easing
    - MagneticButton: Buttons attracted to cursor
    - GlowButton: Pulse glow effects
    - RippleButton: Material Design ripple
    - FloatingElement: Continuous floating animation
    - ParallaxLayer: Mouse-following parallax
- ☑ **CommandPalette.tsx** - Keyboard navigation (Ctrl+K)
    - Fuzzy search across all pages
    - Actions: Navigate, Connect Wallet, Toggle Theme
    - Arrow key navigation
- ☑ **Advanced CSS Animations**
    - Glassmorphism 2.0 with multi-layer blur
    - Holographic shine gradients
    - Depth shadows for 3D appearance
    - Stagger-fade for list items
    - Animated gradient text
    - Reduced motion accessibility support
- ☑ **All Pages Enhanced**
    - Homepage: TiltCard on SwapWidget and stats
    - Pools: Glass-premium cards with gradient icons
    - Vault: Holographic effects on stats cards
    - Bridge: Animated backgrounds and TiltCard

Phase 5: Documentation & Launch Prep ✅ COMPLETE

5.1 Documentation

- ☑ **README.md** - Quick start, architecture overview
- ☑ **XYLONET_PROJECT.md** - Detailed project documentation
- ☑ **contracts/README.md** - Smart contract docs with examples
- ☑ **SECURITY.md** - Security considerations
- ☑ **CONTRIBUTING.md** - Contribution guidelines
- ☑ **LICENSE** - MIT License

5.2 Code Quality

- ☑ Frontend builds successfully
- ☑ No TypeScript errors
- ☑ All contracts verified on Blockscout
- ☑ ABIs properly exported

- ☑ Constants file complete

5.3 Deployment

- ☑ Frontend deployed to Vercel
- ☑ Custom domain configured (xylonet.xyz)
- ☑ Vercel Analytics integrated
- ☑ Speed Insights enabled
- ☑ GitHub repository public

5.4 Git Repository

- ☑ Clean git history
- ☑ All features pushed to main branch
- ☑ Repository: https://github.com/Panchu11/XyloNet

PART B: PROPOSED ENHANCEMENTS

Phase 6: Core Arc Integrations 🔄 PROPOSED

Duration: 4-5 weeks
Priority: CRITICAL
Arc Builders Fund Alignment: FX, Enterprise Ready

6.1 StableFX RFQ Integration

Estimated Time: 2 weeks
Dependencies: Circle API access, KYB verification

6.1.1 Backend Tasks

- □ Register for StableFX API access with Circle
- □ Complete KYB/AML verification process
- □ Implement quote request service
- □ Implement trade creation flow
- □ Implement EIP-712 signing for trade intent
- □ Implement EIP-712 signing for Permit2 funding
- □ Create webhook handler for trade status updates
- □ Add rate comparison logic (AMM vs RFQ)

6.1.2 Smart Contract Tasks

- □ Create `XyloRFQRouter.sol` for RFQ execution
- □ Integrate Permit2 for funding authorization
- □ Add FxEscrow interaction functions
- □ Deploy and verify on Arc Testnet

6.1.3 Frontend Tasks

- □ Create `RFQWidget.tsx` component

- □ Add "Best Rate" toggle (AMM vs RFQ)
- □ Implement quote comparison UI
- □ Add EIP-712 signature flow UI
- □ Create step-by-step RFQ progress indicator

6.1.4 Code Example

```
// StableFX Quote Request
POST /v1/exchange/stablefx/quotes
{
  "from": { "currency": "USDC", "amount": "10000.00" },
  "to": { "currency": "EURC" },
  "tenor": "instant"
}

// Response
{
  "id": "uuid-quote",
  "rate": "0.9150",
  "from": { "currency": "USDC", "amount": "10000.00" },
  "to": { "currency": "EURC", "amount": "9150.00" },
  "expiry": "2025-08-07T11:05:00Z"
}
```

6.1.5 Acceptance Criteria

```
✓ Users can request RFQ quotes for USDC ↔ EURC trades
✓ RFQ automatically suggested for trades >$10,000
✓ Rate comparison shows AMM vs RFQ pricing
✓ Trades settle via FxEscrow contract
```

6.2 Account Abstraction (Gasless Transactions)

Estimated Time: 2 weeks
Dependencies: Pimlico/ZeroDev setup

6.2.1 Infrastructure Tasks

- □ Register for Pimlico bundler API
- □ Configure paymaster for sponsored transactions
- □ Set up gas sponsorship policies
- □ Integrate ZeroDev SDK for smart accounts
- □ Configure session key policies

6.2.2 Frontend Tasks

- □ Add smart wallet connection option
- □ Create "Gasless Mode" toggle
- □ Implement social login (Google, Apple)
- □ Add session key management UI

- □ Show gas savings indicator

6.2.3 Code Example

```
import { createSmartAccountClient } from "permissionless";
import { createPimlicoClient } from "permissionless/clients/pimlico";

const pimlicoClient = createPimlicoClient({
  transport: http("https://api.pimlico.io/v2/arc-testnet/rpc"),
});

const smartAccountClient = createSmartAccountClient({
  account: simpleAccount,
  chain: arcTestnet,
  paymaster: pimlicoClient,
});

// Gasless swap - user pays nothing!
const txHash = await smartAccountClient.sendTransaction({
  to: ROUTER_ADDRESS,
  data: encodeFunctionData({
    abi: XYLO_ROUTER_ABI,
    functionName: "swap",
    args: [swapParams],
  }),
});
```

6.2.4 Acceptance Criteria

```
✓ Users can create smart account wallets
✓ Swaps execute without user paying gas
✓ Social login works (Google, Apple)
✓ Session keys enable automated trading
```

6.3 Permit2 Gasless Approvals

Estimated Time: 1 week

6.3.1 Smart Contract Tasks

- □ Update `XyloRouter.sol` to accept Permit2 signatures
- □ Add `swapWithPermit()` function
- □ Add `addLiquidityWithPermit()` function
- □ Deploy updated router

6.3.2 Frontend Tasks

- □ Detect if user has Permit2 approval
- □ Prompt one-time Permit2 approval if not
- □ Generate EIP-712 permit signatures for swaps
- □ Show "Sign once, swap forever" messaging

6.3.3 Code Example

```
// XyloRouter.sol addition
function swapWithPermit(
    SwapParams calldata params,
    IPermit2.PermitSingle calldata permit,
    bytes calldata signature
) external returns (uint256 amountOut) {
    PERMIT2.permit(msg.sender, permit, signature);
    PERMIT2.transferFrom(msg.sender, address(this), uint160(params.amountIn),
params.tokenIn);
    return _swap(params);
}
```

6.4 Gateway Integration

Estimated Time: 1 week

6.4.1 Tasks

- □ Register for Gateway API access
- □ Implement deposit tracking
- □ Implement cross-chain balance queries
- □ Add "Unified Balance" view to UI
- □ Enable instant cross-chain swaps

6.4.2 Acceptance Criteria

```
✓ Users see unified USDC balance across chains
✓ Instant access to funds on any supported chain
```

Phase 7: Capital Formation Features 🔄 PROPOSED

Duration: 5-6 weeks
Priority: HIGH
Arc Builders Fund Alignment: Markets, Agentic Commerce

7.1 Prediction Markets

Estimated Time: 3 weeks

7.1.1 Smart Contract Tasks

XyloPredictionFactory.sol

- □ Market creation function
- □ Fee configuration
- □ Oracle whitelist management
- □ Emergency pause capability

XyloPredictionMarket.sol

- □ Binary outcome handling (YES/NO)

- □ Share minting/burning
- □ USDC collateralization
- □ Resolution via oracle
- □ Redemption after resolution

```
contract XyloPredictionMarket {
    IERC20 public immutable usdc;

    function buyYes(uint256 usdcAmount) external {
        usdc.transferFrom(msg.sender, address(this), usdcAmount);
        uint256 shares = calculateShares(usdcAmount, true);
        yesBalances[msg.sender] += shares;
    }

    function buyNo(uint256 usdcAmount) external {
        usdc.transferFrom(msg.sender, address(this), usdcAmount);
        uint256 shares = calculateShares(usdcAmount, false);
        noBalances[msg.sender] += shares;
    }

    function resolve(Outcome _outcome) external onlyOracle {
        outcome = _outcome;
    }

    function redeem() external {
        uint256 payout = calculatePayout(msg.sender);
        usdc.transfer(msg.sender, payout);
    }
}
```

7.1.2 Oracle Integration

- □ Integrate Pyth Network for price feeds
- □ Integrate Chainlink for general data
- □ Create oracle adapter interface

7.1.3 Frontend Tasks

- □ Market browser with categories
- □ Market trading interface (buy YES/NO)
- □ Market creation form
- □ Portfolio view
- □ Resolution tracker

7.2 AI Agent Interface (MCP Tools)

Estimated Time: 1 week

7.2.1 API Endpoints

Endpoint	Method	Description
`/api/v1/quote`	GET	Get swap quote
`/api/v1/swap`	POST	Execute swap
`/api/v1/bridge`	POST	Bridge USDC
`/api/v1/pools`	GET	Get pool stats
`/api/v1/vault/deposit`	POST	Deposit to vault
`/api/v1/vault/withdraw`	POST	Withdraw from vault

7.2.2 MCP Tool Definition

```
{
  "name": "xylonet",
  "version": "1.0.0",
  "tools": [
    {
      "name": "get_swap_quote",
      "description": "Get a price quote for swapping stablecoins",
      "parameters": {
        "tokenIn": { "type": "string", "enum": ["USDC", "EURC", "USYC"] },
        "tokenOut": { "type": "string", "enum": ["USDC", "EURC", "USYC"] },
        "amountIn": { "type": "number" }
      }
    },
    {
      "name": "execute_swap",
      "description": "Execute a stablecoin swap"
    },
    {
      "name": "bridge_usdc",
      "description": "Bridge USDC to another chain"
    },
    {
      "name": "deposit_vault",
      "description": "Deposit USDC into vault for yield"
    }
  ]
}
```

7.2.3 Automated Strategies

- □ DCA (Dollar Cost Averaging)
- □ Portfolio Rebalancing
- □ Yield Optimization
- □ Arbitrage Detection

7.3 Limit Orders & Advanced Trading

Estimated Time: 2 weeks

7.3.1 Order Types

Order Type	Description
Limit	Execute at specific price
TWAP	Time-weighted over duration
Stop-Loss	Execute when price falls below threshold
Take-Profit	Execute when price rises above threshold

7.3.2 Smart Contract Tasks

- □ Create `XyloLimitOrder.sol`
- □ Create `XyloTWAP.sol`
- □ Integrate keeper network (Gelato/Chainlink)

7.3.3 Frontend Tasks

- □ Limit order form
- □ TWAP configuration interface
- □ Active orders list
- □ Order history

Phase 8: Advanced Optimizations 🔄 PROPOSED

Duration: 3-4 weeks
Priority: MEDIUM

8.1 USYC Yield Strategies

Estimated Time: 2 weeks

8.1.1 Strategy: Auto-Compound USYC

```
contract XyloUSYCVault is ERC4626 {
    uint256 public usycAllocation = 80; // 80% in USYC

    function deposit(uint256 assets, address receiver) public override returns (uint256) {
        usdc.transferFrom(msg.sender, address(this), assets);

        // Swap portion to USYC
        uint256 usycAmount = assets * usycAllocation / 100;
        router.swap(usdc, usyc, usycAmount);

        return _mint(receiver, convertToShares(assets));
    }
    function totalAssets() public view override returns (uint256) {
        return usdc.balanceOf(address(this)) +
               router.getAmountOut(usyc, usdc, usyc.balanceOf(address(this)));
    }
}
```

8.2 Multi-hop Routing

Estimated Time: 1 week

8.2.1 Tasks

- □ Implement path finding algorithm
- □ Multicall3 integration for gas efficiency
- □ Split routing for large trades

8.3 Analytics Dashboard

Estimated Time: 2 weeks

8.3.1 Metrics

Metric	Description
Protocol TVL	Total value locked across pools
24h Volume	Trading volume in USD
24h Fees	Protocol fees generated
Active Users	Unique addresses per day
Bridge Volume	CCTP transfer volume by chain
Vault APY	Historical yield performance

Phase 9: Governance & Token 🔄 PROPOSED

Duration: 4-5 weeks
Priority: Future

9.1 Token Contracts

XyloToken.sol

- □ ERC20 with 18 decimals
- □ Initial supply: 100,000,000 XYLO
- □ Minting capability (capped)
- □ Burning capability

XyloStaking.sol

- □ Stake XYLO to earn protocol fees
- □ Lock periods for bonus rewards

- ☐ Fee distribution mechanism

XyloGovernor.sol

- ☐ OpenZeppelin Governor framework
- ☐ Proposal creation with threshold
- ☐ Voting with staked XYLO
- ☐ Timelock execution

9.2 Tokenomics

Category	Allocation	Vesting
Community & Ecosystem	40%	4 years linear
Team & Advisors	20%	1 year cliff, 3 years
Investors	15%	6 month cliff, 2 years
Liquidity Mining	15%	3 years emissions
Treasury	10%	DAO controlled

PART C: TECHNICAL REFERENCE

Deployed Contract Addresses

XyloNet Contracts (Arc Testnet)

Contract	Address	Verified
XyloFactory	`0x60EDeFB094B84BBC6430cc130B358A43Ba1979e2`	✅
XyloRouter	`0x73742278c31a76dBb0D2587d03ef92E6E2141023`	✅
XyloBridge (V2)	`0xf7Df65Ce418E938ee8d9a0A0d227A43441fe4641`	✅
XyloVault	`0x240Eb85458CD41361bd8C3773253a1D78054f747`	✅
USDC-EURC Pool	`0x3DF3966F5138143dce7a9cFDdC2c0310ce083BB1`	✅
USDC-USYC Pool	`0x8296cC7477A9CD12cF632042fDDc2aB89151bb61`	✅

Arc Network Native Contracts

Contract	Address	Purpose
USDC (Native)	`0x3600000000000000000000000000000000000000`	Native stablecoin
EURC	`0x89B50855Aa3bE2F677cD6303Cec089B5F319D72a`	Euro stablecoin
USYC	`0xe9185F0c5F296Ed1797AaE4238D26CCaBEadb86C`	Yield-bearing token

TokenMessengerV2	`0x8FE6B999Dc680CcFDD5Bf7EB0974218be2542DAA`	CCTP V2 messenger
MessageTransmitterV2	`0xE737e5cEBEEBa77EFE34D4aa090756590b1CE275`	CCTP V2 transmitter
Gateway Wallet	`0x0077777d7EBA4688BDeF3E311b846F25870A19B9`	Chain abstraction
FxEscrow	`0x1f91886C7028986aD885ffCee0e40b75C9cd5aC1`	StableFX settlement
Permit2	`0x000000000022D473030F116dDEE9F6B43aC78BA3`	Gasless approvals
Multicall3	`0xcA11bde05977b3631167028862bE2a173976CA11`	Batch calls

Architecture Diagrams

Smart Contract Architecture

```
contracts/
├── src/
│   ├── core/
│   │   ├── XyloFactory.sol      ✅ Deployed & Verified
│   │   ├── XyloStablePool.sol   ✅ Deployed & Verified
│   │   ├── XyloRouter.sol       ✅ Deployed & Verified
│   │   └── XyloERC20.sol        ✅ Deployed & Verified
│   ├── bridge/
│   │   └── XyloBridge.sol       ✅ Deployed & Verified (CCTP V2)
│   ├── vault/
│   │   └── XyloVault.sol        ✅ Deployed & Verified
│   ├── orders/                  🔄 Proposed
│   │   ├── XyloLimitOrder.sol
│   │   └── XyloTWAP.sol
│   ├── prediction/              🔄 Proposed
│   │   ├── XyloPredictionFactory.sol
│   │   └── XyloPredictionMarket.sol
│   ├── governance/              🔄 Proposed
│   │   ├── XyloToken.sol
│   │   ├── XyloStaking.sol
│   │   └── XyloGovernor.sol
│   └── interfaces/
│       ├── IERC20.sol
│       ├── IXyloFactory.sol
│       ├── IXyloPool.sol
│       └── IXyloRouter.sol
```

Frontend Architecture

```
frontend/
├── src/
│   ├── app/
│   │   ├── page.tsx             ✅ Home / Swap
│   │   ├── bridge/page.tsx      ✅ Bridge
│   │   ├── pools/page.tsx       ✅ Pools
│   │   ├── vault/page.tsx       ✅ Vault
│   │   ├── history/page.tsx     ✅ History
│   │   ├── orders/page.tsx      🔄 Proposed - Limit/TWAP
│   │   ├── predict/page.tsx     🔄 Proposed - Prediction Markets
│   │   ├── analytics/page.tsx   🔄 Proposed - Dashboard
│   │   ├── layout.tsx           ✅ Root layout
│   │   └── globals.css          ✅ Global + Mobile CSS
│   ├── components/
│   │   ├── swap/SwapWidget.tsx  ✅ Token swap
│   │   ├── bridge/BridgeWidget.tsx ✅ Bridge (Circle Kit)
│   │   ├── rfq/RFQWidget.tsx    🔄 Proposed - StableFX
│   │   ├── orders/              🔄 Proposed
│   │   ├── prediction/          🔄 Proposed
│   │   ├── ui/
│   │   │   ├── Toast.tsx        ✅ Notifications
│   │   │   ├── TokenLogos.tsx   ✅ Token + Chain logos
│   │   │   ├── Skeleton.tsx     ✅ Loading states
│   │   │   ├── Tooltip.tsx      ✅ Info tooltips
│   │   │   ├── Confetti.tsx     ✅ Celebrations
│   │   │   ├── EmptyState.tsx   ✅ Empty states
│   │   │   ├── AnimatedBackground.tsx ✅ Particles
│   │   │   ├── TiltCard.tsx     ✅ 3D cards
│   │   │   ├── AnimatedElements.tsx ✅ Interactive
│   │   │   └── CommandPalette.tsx ✅ Keyboard nav
│   │   ├── Header.tsx           ✅ Navigation
│   │   ├── Footer.tsx           ✅ Footer
│   │   └── Providers.tsx        ✅ Web3 providers
│   ├── config/
│   │   ├── wagmi.ts             ✅ Wallet config
│   │   ├── constants.ts         ✅ Contract addresses
│   │   └── abis/index.ts        ✅ Contract ABIs
│   ├── lib/
│   │   ├── utils.ts             ✅ Utilities
│   │   └── transactions.ts      ✅ Tx history
│   └── api/                     🔄 Proposed - Agent API
│       └── v1/
```

Tech Stack Summary

Smart Contracts

Technology	Version	Purpose
Solidity	0.8.30	Contract language
Hardhat	2.x	Development framework
OpenZeppelin	5.x	Security standards

Frontend

Technology	Version	Purpose
Next.js	16	React framework
TypeScript	5.x	Type safety
Tailwind CSS	4.x	Styling
wagmi	2.x	Ethereum hooks
viem	2.x	Ethereum client
RainbowKit	2.x	Wallet UI
Circle Bridge Kit	Latest	Cross-chain UI
Lucide React	Latest	Icons
Vercel Analytics	Latest	Analytics

Infrastructure

Service	Purpose
Arc Testnet	Blockchain network
Circle CCTP V2	Cross-chain messaging
Blockscout	Block explorer
GitHub	Version control
Vercel	Frontend hosting

Smart Contract Specifications

Proposed Contract Summary

Contract	Purpose	Gas Estimate
XyloRFQRouter	StableFX RFQ integration	~200k
XyloLimitOrder	Limit order management	~150k create, 120k execute
XyloTWAP	Time-weighted orders	~180k create
XyloPredictionFactory	Market creation	~250k
XyloPredictionMarket	Market trading	~80k buy, ~60k redeem
XyloUSYCVault	Yield strategy vault	~180k deposit
XyloToken	Governance token	~50k transfer
XyloStaking	Fee distribution	~100k stake

XyloGovernor	DAO governance	~200k propose

API Specifications

Authentication

```
Header: X-API-Key: your_api_key_here
```

Rate Limits

- Free tier: 100 requests/minute
- Pro tier: 1000 requests/minute

Endpoint Summary

Endpoint	Method	Description
`/api/v1/quote`	GET	Get swap quote
`/api/v1/swap`	POST	Execute swap
`/api/v1/bridge`	POST	Bridge USDC
`/api/v1/pools`	GET	List all pools
`/api/v1/pools/{id}`	GET	Get pool details
`/api/v1/vault/deposit`	POST	Deposit to vault
`/api/v1/vault/withdraw`	POST	Withdraw from vault
`/api/v1/vault/stats`	GET	Get vault statistics

Testing & Security

Test Coverage Requirements

Category	Minimum Coverage
Unit Tests	90%
Integration Tests	80%
E2E Tests	70%

Security Checklist

- □ Reentrancy guards on all external calls
- □ CEI pattern (Checks-Effects-Interactions)
- □ Access control for admin functions
- □ Pausability for emergency
- □ Timelocks for sensitive operations

- ☐ Oracle manipulation protection
- ☐ Flash loan attack prevention
- ☐ External security audit

Timeline Summary

```
┌─────────────────────────────────────────────────────────────────────┐
│                    COMPLETE DEVELOPMENT TIMELINE                     │
├─────────────────────────────────────────────────────────────────────┤
│ ✅ COMPLETED PHASES (Phases 1-5)                                    │
├─────────────────────────────────────────────────────────────────────┤
│ Phase 1: Foundation                 │ ✅ Complete                   │
│ Phase 2: Bridge Integration         │ ✅ Complete                   │
│ Phase 3: Yield Features             │ ✅ Complete                   │
│ Phase 4: UX/UI Polish               │ ✅ Complete                   │
│ Phase 5: Documentation              │ ✅ Complete                   │
├─────────────────────────────────────────────────────────────────────┤
│ 🔄 PROPOSED PHASES (Phases 6-9)                                     │
├─────────────────────────────────────────────────────────────────────┤
│ Phase 6: Core Arc Integrations      │ 4-5 weeks                   │
│   - Week 1-2: StableFX RFQ                                        │
│   - Week 2-4: Account Abstraction + Permit2                       │
│   - Week 4-5: Gateway Integration                                 │
├─────────────────────────────────────────────────────────────────────┤
│ Phase 7: Capital Formation          │ 5-6 weeks                   │
│   - Week 5-8: Prediction Markets                                  │
│   - Week 8-9: AI Agent Tools (MCP)                                │
│   - Week 9-11: Limit Orders + TWAP                                │
├─────────────────────────────────────────────────────────────────────┤
│ Phase 8: Advanced Optimizations     │ 3-4 weeks                   │
│   - Week 11-13: USYC Strategies + Multi-hop                       │
│   - Week 13-15: Analytics Dashboard                               │
├─────────────────────────────────────────────────────────────────────┤
│ Phase 9: Governance & Token         │ 4-5 weeks                   │
│   - Week 15-20: Token + Staking + DAO (if proceeding)             │
└─────────────────────────────────────────────────────────────────────┘
```

Current Features Summary

Feature	Description	Status
Swap	Trade USDC ↔ EURC ↔ USYC with StableSwap curve	✅ Live
Pools	Add/remove liquidity, view reserves, earn fees	✅ Live
Bridge	Cross-chain USDC via Circle CCTP V2 (7 chains)	✅ Live
Vault	Deposit USDC to earn yield (ERC-4626)	✅ Live
History	View past transactions with filters	✅ Live

Mobile	Fully responsive design for all screens	✅ Live
Immersive UX	Particles, 3D cards, glassmorphism, animations	✅ Live
Command Palette	Keyboard navigation with Ctrl+K	✅ Live
StableFX RFQ	Institutional FX trading	🔄 Proposed
Account Abstraction	Gasless transactions	🔄 Proposed
Prediction Markets	Binary outcome markets	🔄 Proposed
AI Agent Tools	MCP definitions for automation	🔄 Proposed
Limit Orders	Advanced trading features	🔄 Proposed
Governance	XYLO token + DAO	🔄 Proposed

Explorer Links

Contract	Explorer
Factory	https://testnet.arcscan.app/address/0x60EDeFB094B84BBC6430cc130B358A43Ba1979e2
Router	https://testnet.arcscan.app/address/0x73742278c31a76dBb0D2587d03ef92E6E2141023
Bridge	https://testnet.arcscan.app/address/0xf7Df65Ce418E938ee8d9a0A0d227A43441fe4641
Vault	https://testnet.arcscan.app/address/0x240Eb85458CD41361bd8C3773253a1D78054f747
USDC-EURC Pool	https://testnet.arcscan.app/address/0x3DF3966F5138143dce7a9cFDdC2c0310ce083BB1
USDC-USYC Pool	https://testnet.arcscan.app/address/0x8296cC7477A9CD12cF632042fDDc2aB89151bb61

Resources

Resource	Link
GitHub Repository	https://github.com/Panchu11/XyloNet
Live Application	https://xylonet.xyz
Arc Network Docs	https://docs.arc.network
StableFX Docs	https://developers.circle.com/stablefx
Gateway Docs	https://developers.circle.com/gateway
Pimlico Docs	https://docs.pimlico.io
ZeroDev Docs	https://docs.zerodev.app
Arc Builders Fund	https://www.circle.com/blog/introducing-the-arc-builders-fund

---

XyloNet - Complete Implementation Plan & Enhancement Roadmap

Built for the Arc Network Ecosystem

Prepared for Arc Builders Fund Application