ZK Proof

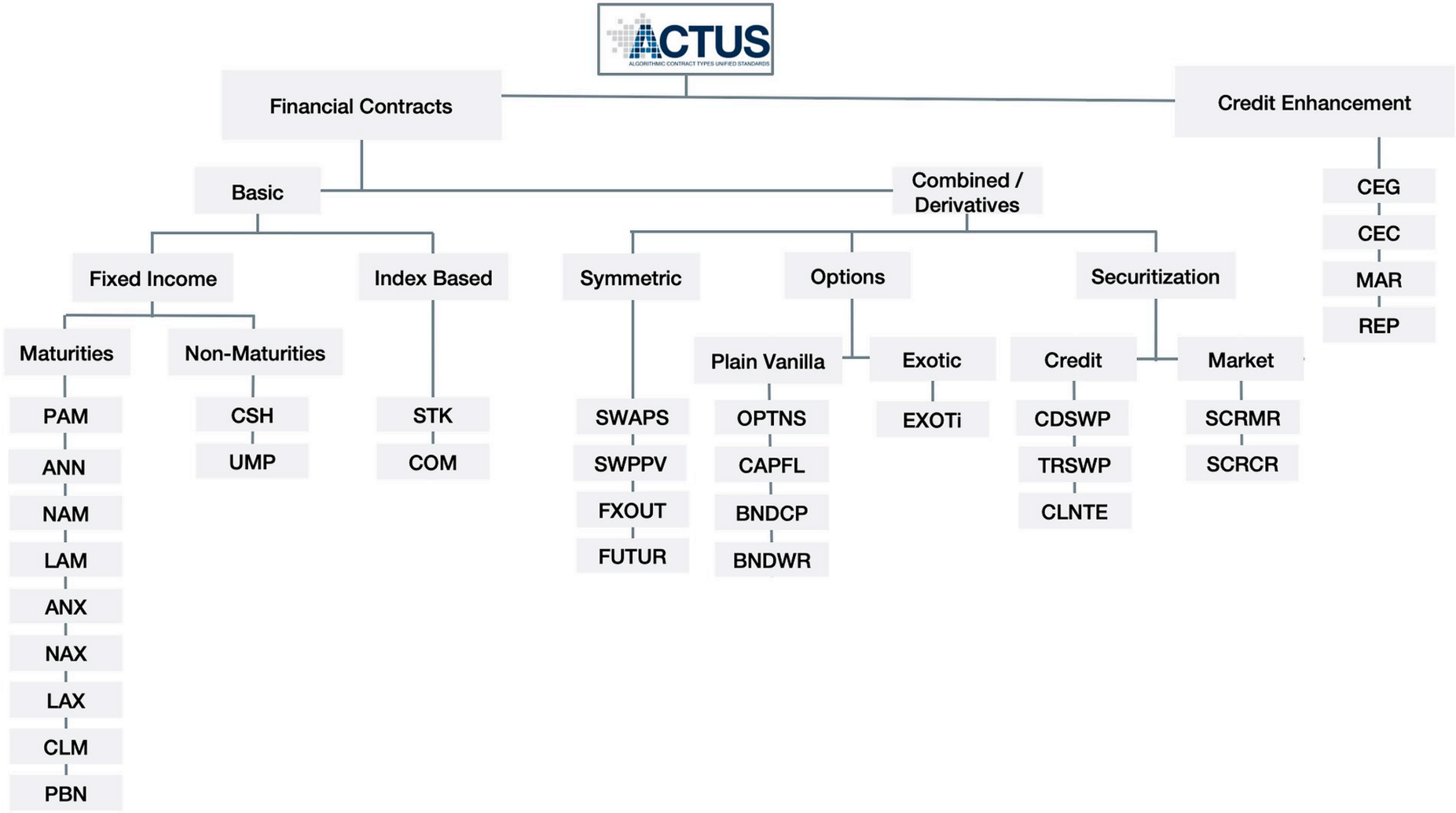
6th International Workshop

ZK-ACTUS

Verifiable Financial Contracts

Part 1: ACTUS

An Emerging Financial Standard


ACTUS
ALGORITHMIC CONTRACT TYPES UNIFIED STANDARDS
Financial Contracts
Credit Enhancement
Basic
Combined / Derivatives
CEG
CEC
MAR
REP
Fixed Income
Index Based
Symmetric
Options
Securitization
Maturities
Non-Maturities
Plain Vanilla
Exotic
Credit
Market
PAM
ANN
NAM
LAM
ANX
NAX
LAX
CLM
PBN
CSH
UMP
STK
COM
SWAPS
SWPPV
FXOUT
FUTUR
OPTNS
CAPFL
BNDCP
BNDWR
EXOTi
CDSWP
TRSWP
CLNTE
SCRMR
SCRCR

Dictionary

Taxonomy

Contract

Term Set

Term

Applicability

Enum

Scalar Type

Function Type

Algorithms

- **Types**
  - Utility Functions
  - State Transition Functions
  - Payoff Functions
- **Inputs**
  - Machine readable termsets
  - Terms are composable
  - Hetereogenous
- **Output**
  - Event Sequence (1..N)
  - Equivalent to cash flows
  - Homogeneous

ACTUS

Implementations

JAVA
(reference)

RUST
(WIP)

HASKELL

PYTHON
(WIP)

TYPESCRIPT
(WIP)

SOLIDITY

Part 2: ACTUS + ZK + DLT

Verifiable Financial Contracts

ACTUS + ZK + DLT

VFC Pillars

VFC Integrity

VFC Tokenisation

VFC Payments

ACTUS + ZK + DLT

ACTUS

(Counter Parties, Term Set, Algorithm, Cash Flows)

Cryptographic Proofs

(Signatures, Attestations, Fingerprints, ZK-Proofs)

DLT

(Smart Contract)

Minting

(Identifiers, Direction, Counter Parties, Units, Metadata)

DLT

(Smart Contract)

Servicing

(Auditors, Rating, Regulators, Markets)

ACTUS + ZK + DLT

ACTUS Cash Flow

(Timestamp, Direction, Amount, Denomination, Obligor)

Payments Engine

(Verify, Calculate, Open, Close, Default, Notify)

DLT

(Smart Contract)

VFC Principles

- **Occams Razor**
  As Little As Possible, As Much As Necessary
- **Chain Agnostic**
  Standard Smart Contracts
- **Privacy Preserving**
  Who, What, When, Why
- **Trust But Verify**
  Cryptographic Proofs Everywhere

VFC Challenges

- **Regulatory Certitude**
  Robust. Nuanced. Adaptive.
- **Counter-Party Risk**
  Identity -> KYC/AML. Defaults -> ???
- **Post Quantum Security**
  Cryptography equivalent to Y2K
- **Jurisdictional Anchoring**
  Smart Legal Contracts
- **Technological Flux**
  Multi-Decadal Platforms

Part 3: ACTUS ZK Proofs

Computational Integrity

ACTUS + ZK + DLT

ACTUS

(Counter Parties, Term Set, Algorithm, Cash Flows)

Cryptographic Proofs

(Signatures, Attestations, Fingerprints, ZK-Proofs)

DLT

(Smart Contract)

ZK-Proofs

f(x,w) → {True,False}

Properties

Succint
Sound
Expensive to compute
Cheap to verify

Elements

Arithmetic Circuit
Constraint System
Polynomial
Polynomial Commitment

Developers

Virtual Machines
E-DSLs
Rollups
Applications

Part 4: ACTUS Gateway

L2 <-> L1 Infrastructure

Introduction

Leveraging DLT to service ACTUS compliant financial contracts is the R&D team's focalising use case. It is an activity well suited to a tightening regulatory environment in which 'crypto' is deemed a regulated activity.

ACTUS algorithms must be formally and operationally verifiable. In respect of operational verifiability, the R&D team is building a special purpose ZK infrastructure to service ACTUS financial contracts at scale.

DLT will be used to publish set of cryptographic proofs encompassing the entire lifecycle of a financial contract. Such proofs include standard constructs such as data fingerprints (i.e. hashes) as well as ZK proofs pertaining to the verifiably correct execution of ACTUS algorithms.

The bedrock of published proofs represents an integrity layer upon which tokenisation & payment systems may be established.

ACTUS Gateway

Solution Elements

API Gateway

ZK Provers

Data Availability

DLT Contracts

ZK Proof

6th International Workshop

ZK-ACTUS

Verifiable Financial Contracts