IOTA
EVANGELIST
NETWORK

Streams Workshop

What they are and how to use them

Alessandro Buser

May 2020

Agenda

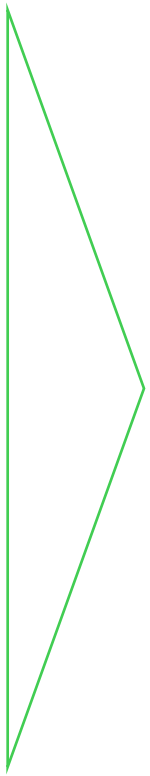
We will have a 45 min. workshop with an initial theoretical part and then some coding

1	From MAM to Streams	‘5
2	Stream Applications	‘5
3	The Channels Application	‘5
4	Linking Messages in Channels	‘5
5	The Messaging Flow in Channels	‘5
6	Coding :)	‘20

IOTA-Streams is an abstraction and generalization of MAM

The learnings taken from MAM have been used to create a more generally applicable solution

MAMv0	Streams
Released in **Nov. 2017**	Released in **Feb. 2020**
Initial libraries in: ▪ Python ▪ JavaScript/Node.js ▪ Rust	Initial libraries in: ▪ Rust ▪ C
Goals: Provide a solution to send encrypted, unalterable and authored data using the Tangle	**Goals**: Provide a framework for cryptographic Protocols and Applications
Supports 3 modes: - **Public** (Broadcast) - **Private** (Encrypted) - **Restricted** (Encrypted + Revocable)	Supports **many Transport modes**, for e.g. Tangle, HTTP, TCP Supports **many types of applications**, MAM Channels is just one of them Better Subscriber Management



IOTA-Streams Can be Used to Create Many Applications

The Underlying Cryptographic methods can be leveraged to enable many different usecases

The Streams Stack

- **Core layers** utils, Troika, keys, p-rng
- **Keccak** Keccak-F[1600] as spongos transform
- **Traversable Merkle tree**
- **Merkle** signatures over Winternitz
- **NTRU** key encapsulation
- **Protobuf3** cryptographic message definition language
- **Application layer** common definitions

For more info see Vlad Semenov IEN Talk

The Application Layer

Contains **Common Patterns** to create:

- Parties
- Roles
- Message Syntax
- Message Semantic

Messages are **encoded using the Protobuf3** syntax

Parties can “wrap” and “unwrap” messages following the given syntax/semantics

This enables the creation of **complex messaging flows** which are “secure by design” as they **leverage the underlying cryptographic methods**

Channels(MAM) is Just One Example of a Streams Application

These is the specification of the Parties, Roles and Messages needed to create Channels

Parties	Roles	
Author	▪ Owner of the channel ▪ Manages Subscribers ▪ Read tagged messages	▪ Send signed messages ▪ Send tagged messages
Subscriber	▪ “Anonymous” member ▪ Verify Authors Messages	▪ Read messages ▪ Send tagged messages

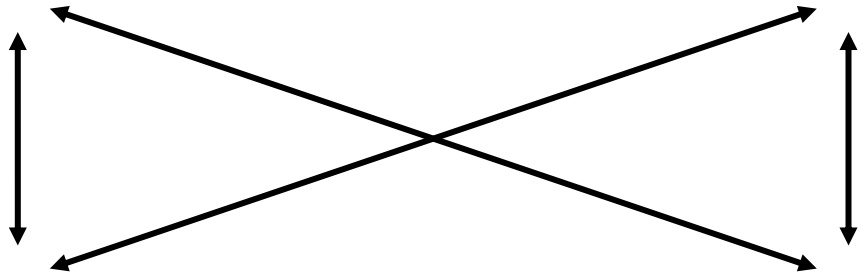
Messages Types		
Announcement Change-Key Keyload	Subscription Un-subscription	Signed Packet Tagged Packet

Compatible with **any** Transport Medium

There are a Total of Four Messaging Modes

They differ in who can write them and who can read them

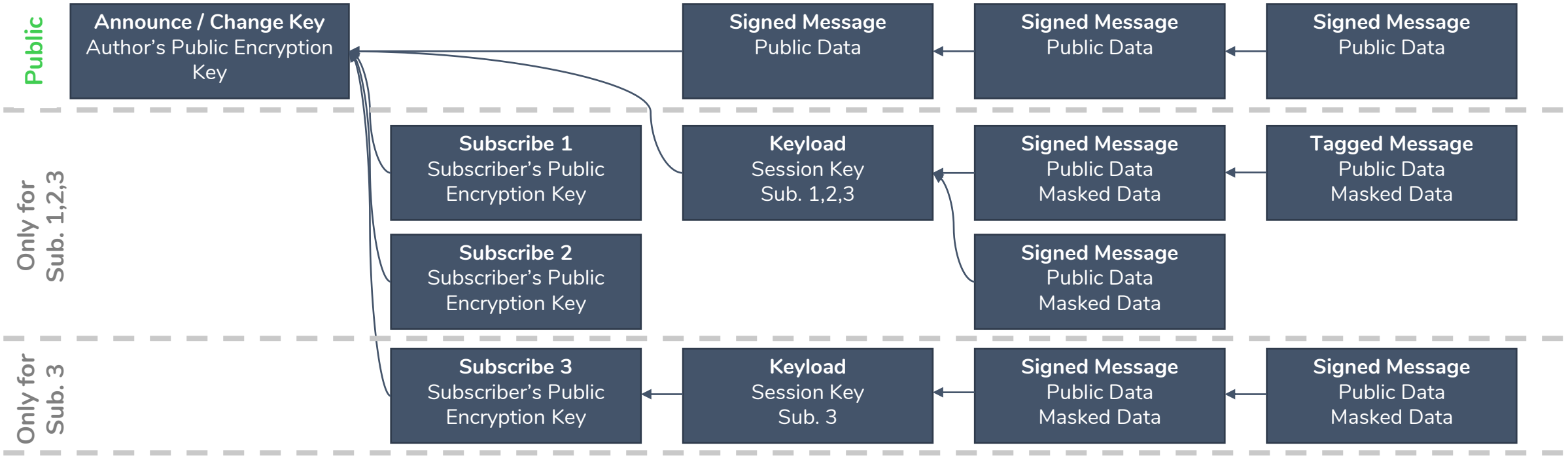
Tagged	Signed
• Both Author and Subscriber can send them • Not possible to identify sender	• Only Author can send and sign them • Origin can be verified



Public	Masked
• Linked to Announce or message after • Anyone can read content	• Linked to Keyload or message after • Only Keyload owners can read

Messages are Linked to Create Dependency Trees

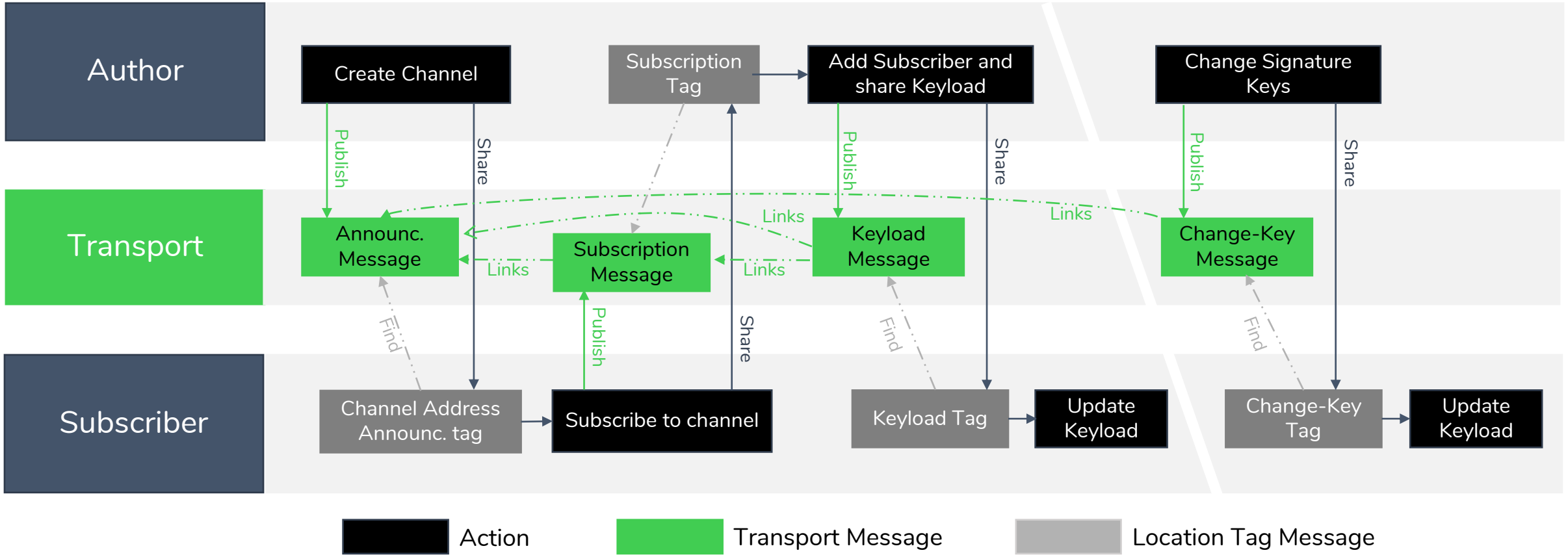
This is to make sure that the Receiver has the information needed to digest the message



- Links can be used to perform more advanced Subscriber Management
- Different Subscribers can have different access to encrypted data
- Masked messages can link to Keyloads indirectly

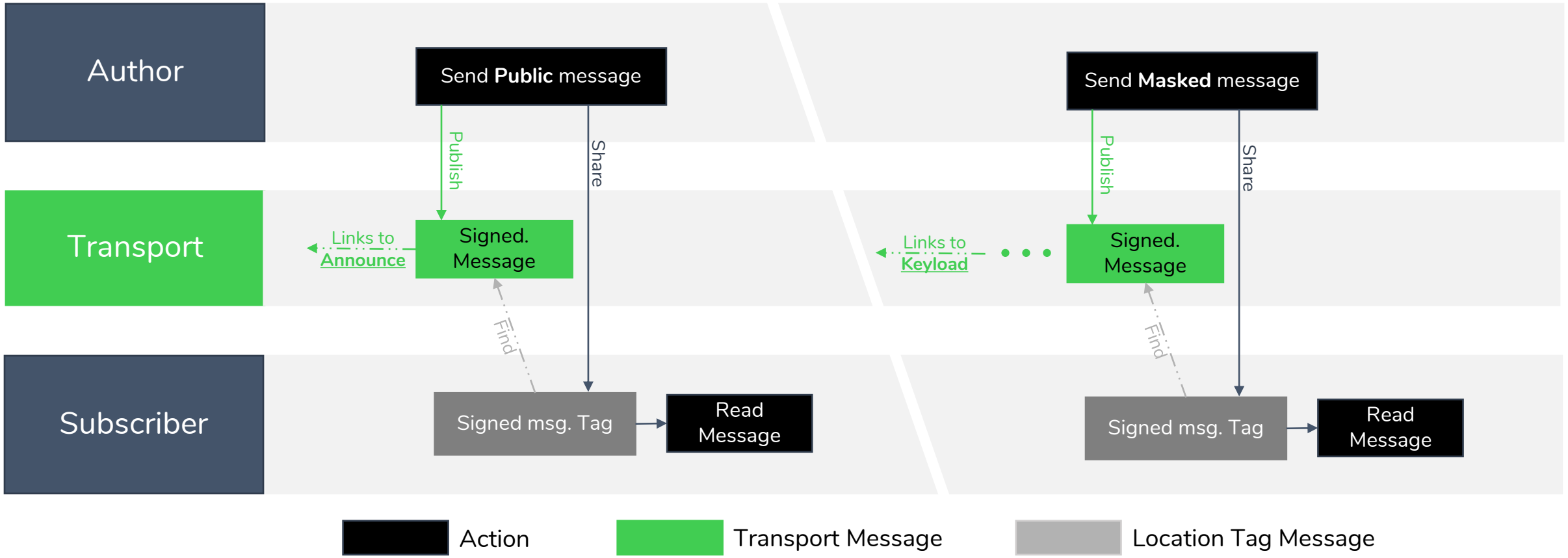
The Message Flows Used in Channels

Adding Subscribers to a Channel



The Message Flows Used in Channels

Sending Public and Masked Messages



How it looks in code

Get your  Rust Ready !

Code Shown Today: https://github.com/AleBuser/IEN-Streams-Workshop

Resources

Development

Rust: https://www.rust-lang.org/tools/install

IOTA-Streams: https://github.com/iotaledger/streams

Channels-Example: https://github.com/JakeSCahill/channels-examples

Channels-Lite: https://github.com/iota-community/iota-channels-lite

Documentation

Rust: https://www.rust-lang.org/learn

IOTA-Streams: https://docs.iota.org/docs/iota-streams/1.0/overview

(In-depth Channels: https://github.com/iotaledger/documentation/pull/712/files**)**

Community

Discord: https://discord.iota.org/

#rust -- #streams-discussion -- #expereince

IOTA EVANGELIST NETWORK

Twitter

@IENofficial

Discord

Discord

Website

ien.io

About the IEN:

IOTA evangelists are people who build a critical mass of support for the IOTA technology, and subsequently establishes IOTA as a technical standard within the global market.
Through the creation of content, demos and PoCs we want to establish IOTA Tangle as the ubiquitous distributed ledger technology (DLT) in the world and the foundational protocol for the upcoming Machine Economy or 4th Industrial Revolution.