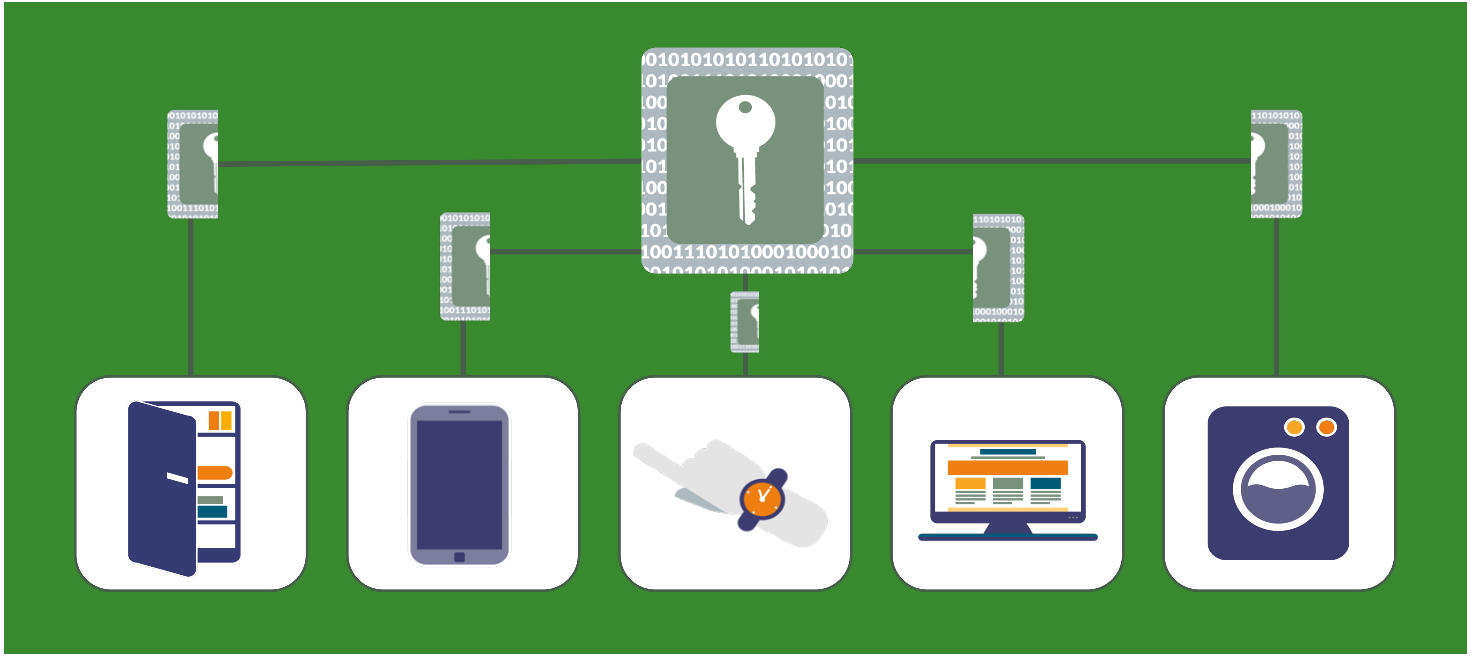
Apache Milagro (incubating)

Brian Spector, Co Founder / CEO, MIRACL
Co Founder, Apache Milagro (incubating)

MILAGRO
milagro.incubator.apache.org

APACHECON
Europe

Apache Milagro: A Distributed Cryptosystem



To Secure the Future of the Web

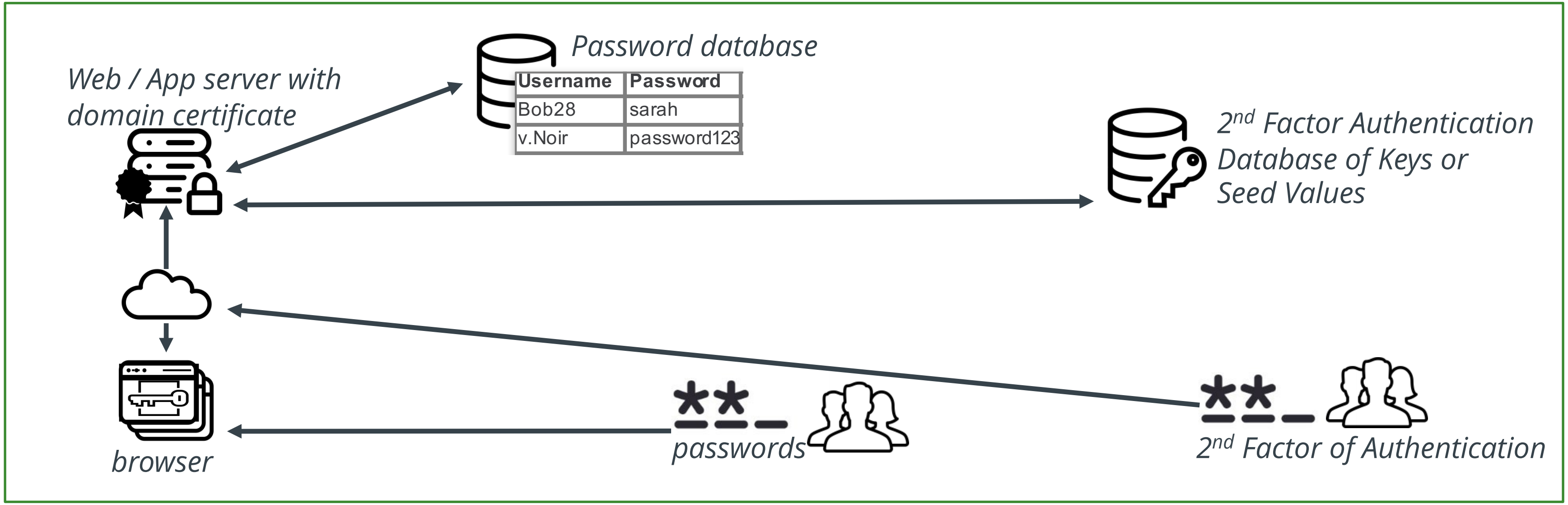
MILAGRO

Centralized Security is PKI / Passwords / 2FA

PKI and Certificates: Mainly used to secure connection between different web browser manufacturers and millions of web sites.

Passwords and / or API Keys: Stored credentials sent from browser / client to back end service to authenticate user or application.

Two-Factor Authentication: Additional authentication deployed in addition to passwords to stop a compromise of account.

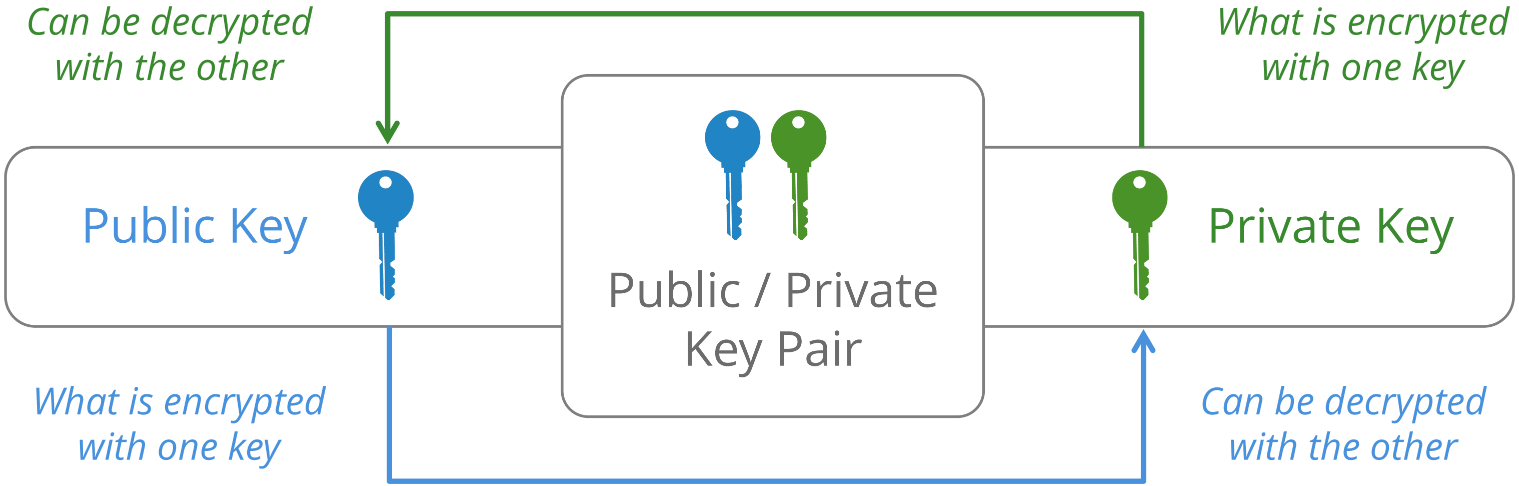


Password database
Web / App server with domain certificate
2nd Factor Authentication Database of Keys or Seed Values
browser
**_
passwords
**_
2nd Factor of Authentication

Username	Password
Bob28	sarah
v.Noir	password123

Public Key Infrastructure (PKI)

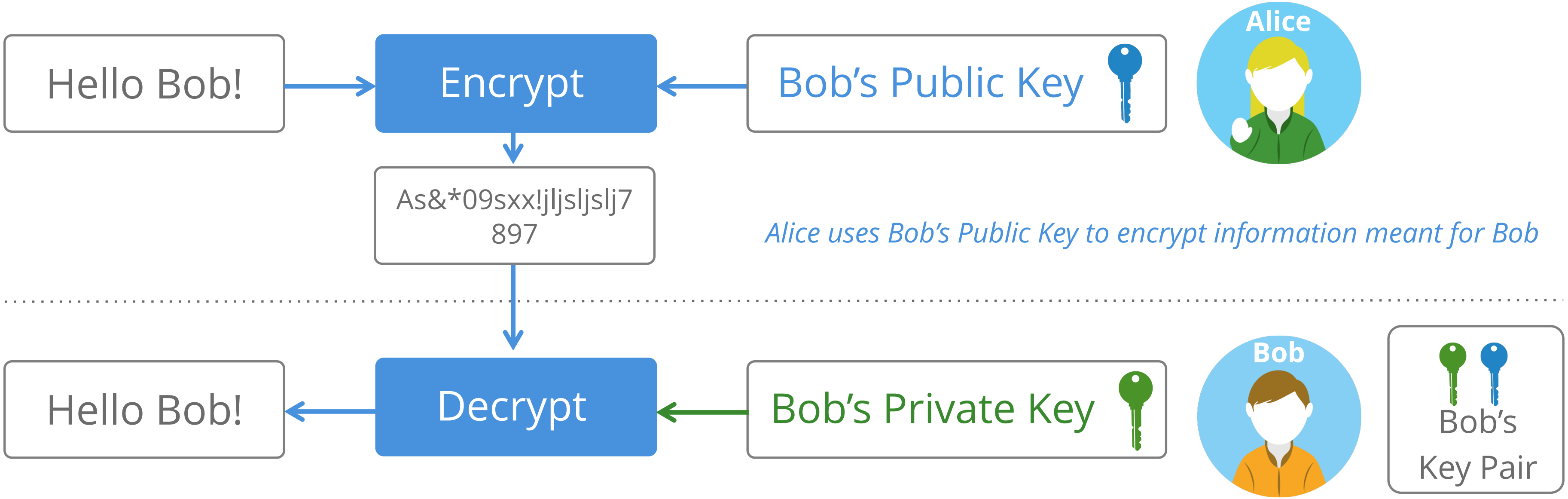
Public / Private Key Refresher:
Current state of the art is Asymmetric Encryption (Public and Private Keys)


Can be decrypted with the other
What is encrypted with one key
Public Key
Public / Private Key Pair
Private Key
What is encrypted with one key
Can be decrypted with the other

Public Key Infrastructure (PKI)

Public / Private Key Refresher

Current state of the art with Alice and Bob example



Hello Bob!

Encrypt

Bob's Public Key

Alice

As&*09sxx!jljsljslj7
897

Alice uses Bob's Public Key to encrypt information meant for Bob

Hello Bob!

Decrypt

Bob's Private Key

Bob

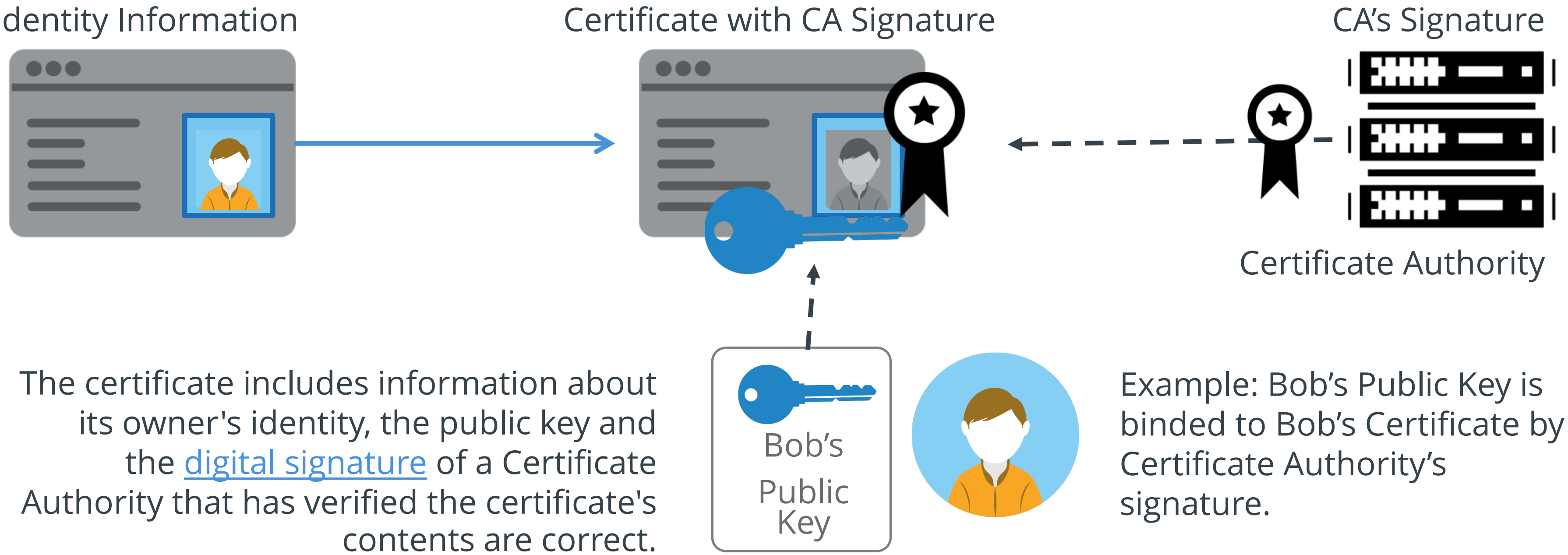
Bob's
Key Pair

Bob uses his Private Key to decrypt information

MILAGRO

Digital Certificates Provide Identity for PKI

An X.509 Digital Certificate is an electronic document used to prove the ownership of a domain, person, app or thing's public key.



Identity Information

Certificate with CA Signature

CA's Signature

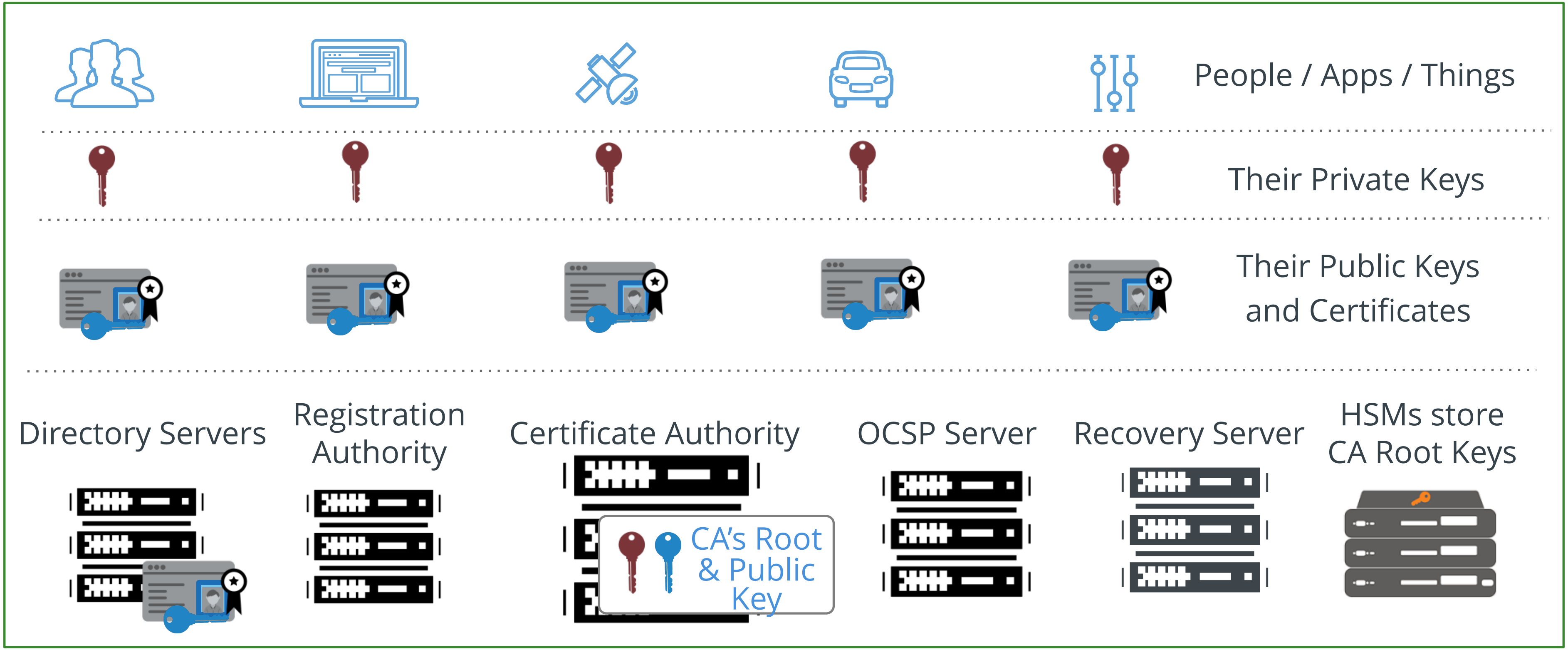
Certificate Authority

Bob's Public Key

The certificate includes information about its owner's identity, the public key and the digital signature of a Certificate Authority that has verified the certificate's contents are correct.

Example: Bob's Public Key is binded to Bob's Certificate by a Certificate Authority's signature.

PKI is Complex, Costly and Vulnerable



People / Apps / Things

Their Private Keys

Their Public Keys and Certificates

Directory Servers

Registration Authority

Certificate Authority

OCSP Server

Recovery Server

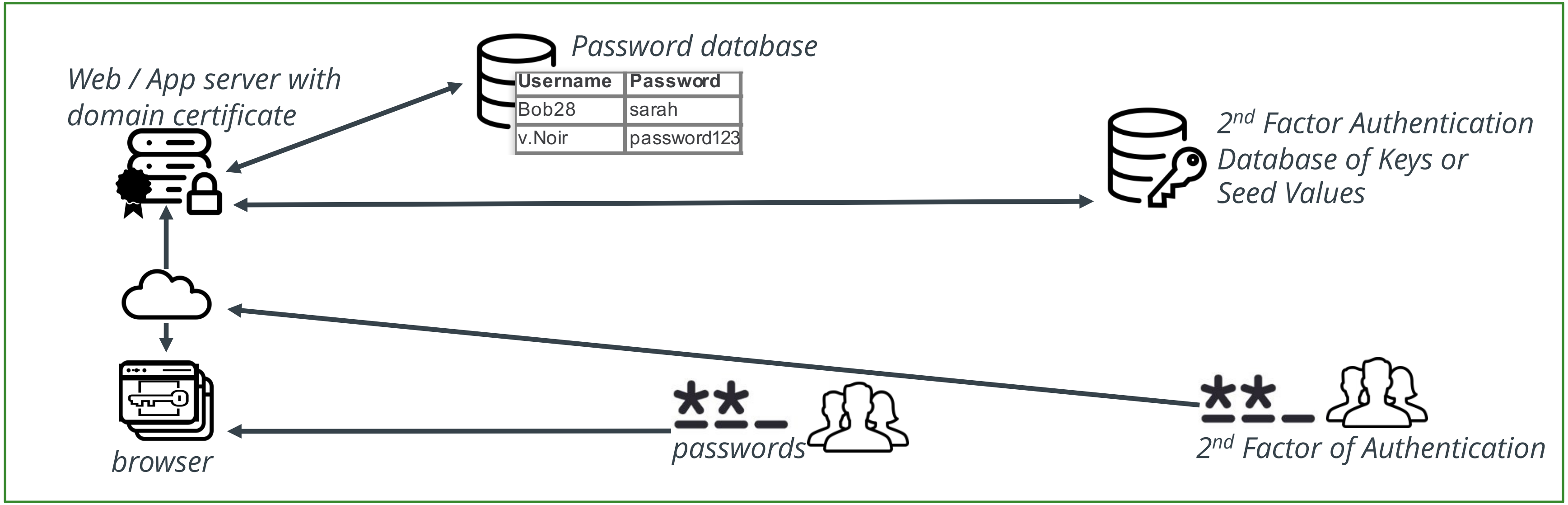
HSMs store CA Root Keys

Centralized Security is PKI / Passwords / 2FA

PKI and Certificates:
Mainly used to secure connection between different web browser manufacturers and millions of web sites.

Passwords and / or API Keys:
Stored credentials sent from browser / client to back end service to authenticate user or application.

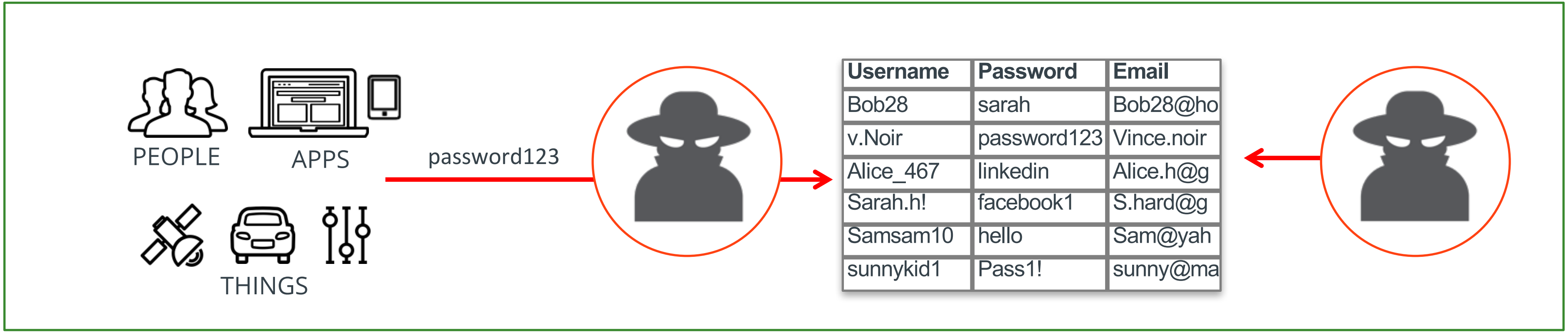
Two-Factor Authentication:
Additional authentication deployed in addition to passwords to stop a compromise of account.


Web / App server with domain certificate
Password database
2nd Factor Authentication Database of Keys or Seed Values
browser
**_
passwords
**_
2nd Factor of Authentication

Username	Password
Bob28	sarah
v.Noir	password123

Centralized Security

Today's Security stores authentication credentials in whole form, in one place, and is easy to compromise.



Username	Password	Email
Bob28	sarah	Bob28@ho
v.Noir	password123	Vince.noir
Alice_467	linkedin	Alice.h@g
Sarah.h!	facebook1	S.hard@g
Samsam10	hello	Sam@yah
sunnykid1	Pass1!	sunny@ma

Most everything on the Internet uses some form of stored credential to authenticate and securely communicate

Credentials sent over the Internet risk being stolen in transit

Since 2013

5 Billion Data Records Breached

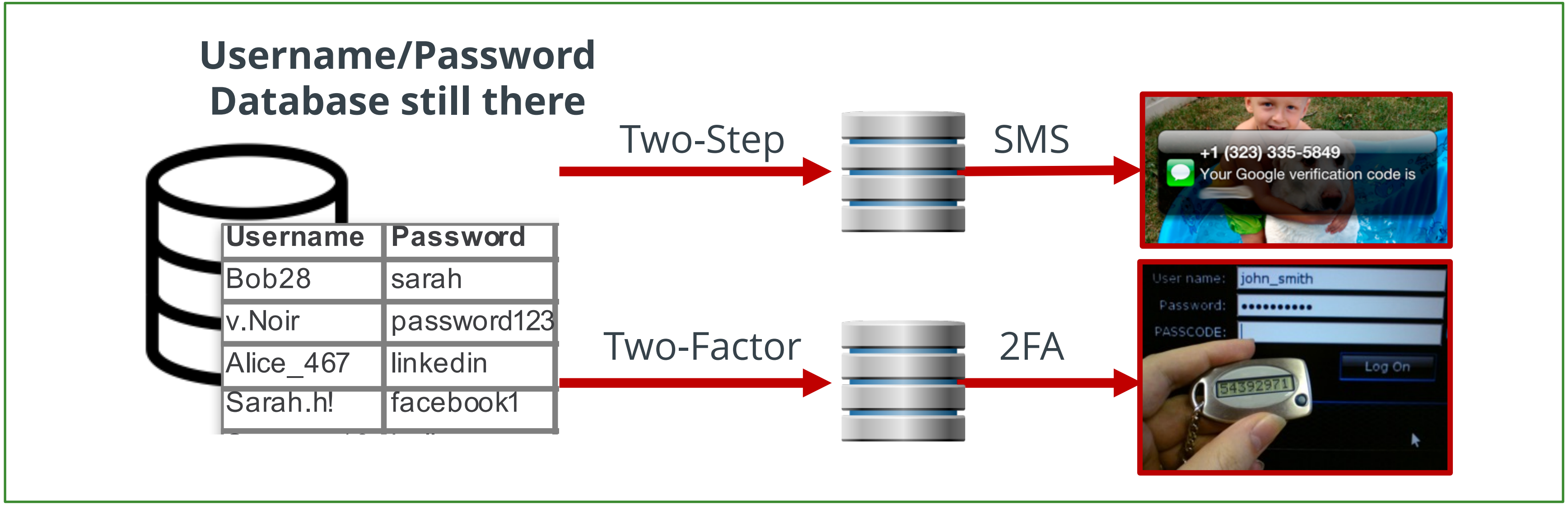
LinkedIn
facebook
YAHOO!
Adobe
EVERNOTE
TARGET

Most everything on the Internet uses some form of stored credential to authenticate and securely communicate

Credentials sent over the Internet risk being stolen in transit

Two-Step and Two-Factor Don’t Remove the Threat

The username / password database still exists, full of passwords, in whole form, in one place, ready to be hacked.



Username/Password Database still there

Username	Password
Bob28	sarah
v.Noir	password123
Alice_467	linkedin
Sarah.h!	facebook1

+1 (323) 335-5849
Your Google verification code is

User name: john_smith
Password: ••••••••••
PASSCODE:
Log On

54392971

Our Ideal Security Architecture

Today

- Centralized (enforced) Trust Authorities
- Single Points of Compromise (root keys)
- x.509 is required because the crypto is old
- Management is hard / revocation is broken
- Proprietary / hard to audit

VS

Tomorrow

- Distributed Cryptosystem with Distributed Trust
- No Single Points of Compromise
- Identity is burned into the keys, no x.509
- Revocation works because less moving parts
- Open source / easily auditable

Apache Milagro: A Distributed Cryptosystem

MILAGRO

milagro.incubator.apache.org

Multi-Factor Authentication and TLS

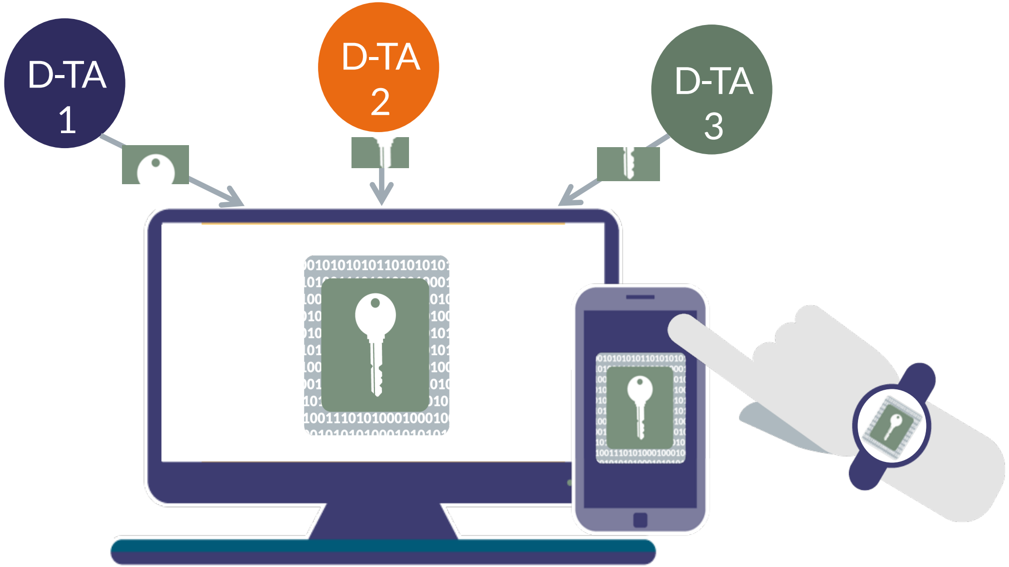
Extend Trust based on your needs

Revoke Trust based on your environments

Scale Trust to Mobile, IoT, and apps

Distributed Trust Ecosystem

Milagro enabled apps and things receive their key shares, or fractions, from Distributed Trust Authorities.



Keys have Identity ”burned in”

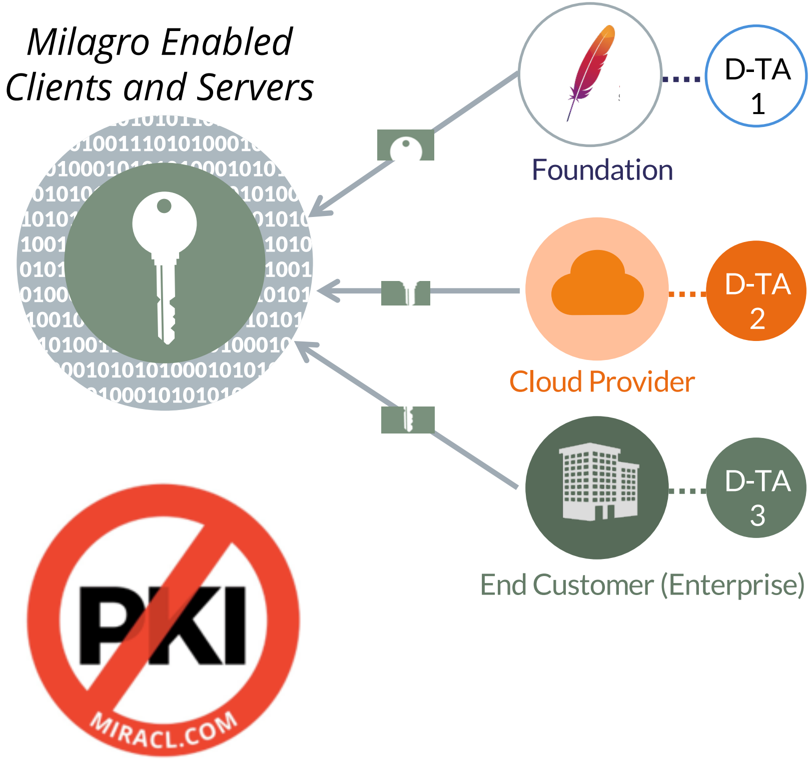
MILAGRO

Distributed Trust Authorities (D-TAS)

Anyone or organization can become a Distributed Trust Authority

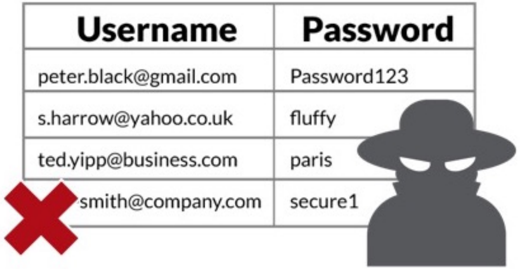
And run it in any geography or jurisdiction

There is no PKI ‘root’ – the future is decentralized


Milagro Enabled Clients and Servers
Foundation
D-TA 1
Cloud Provider
D-TA 2
End Customer (Enterprise)
D-TA 3
PKI
MIRACL.COM

Milagro Multi-Factor Authentication

Eliminates
the risk of password
database breach



Username	Password
peter.black@gmail.com	Password123
s.harrow@yahoo.co.uk	fluffy
ted.yipp@business.com	paris
smith@company.com	secure1

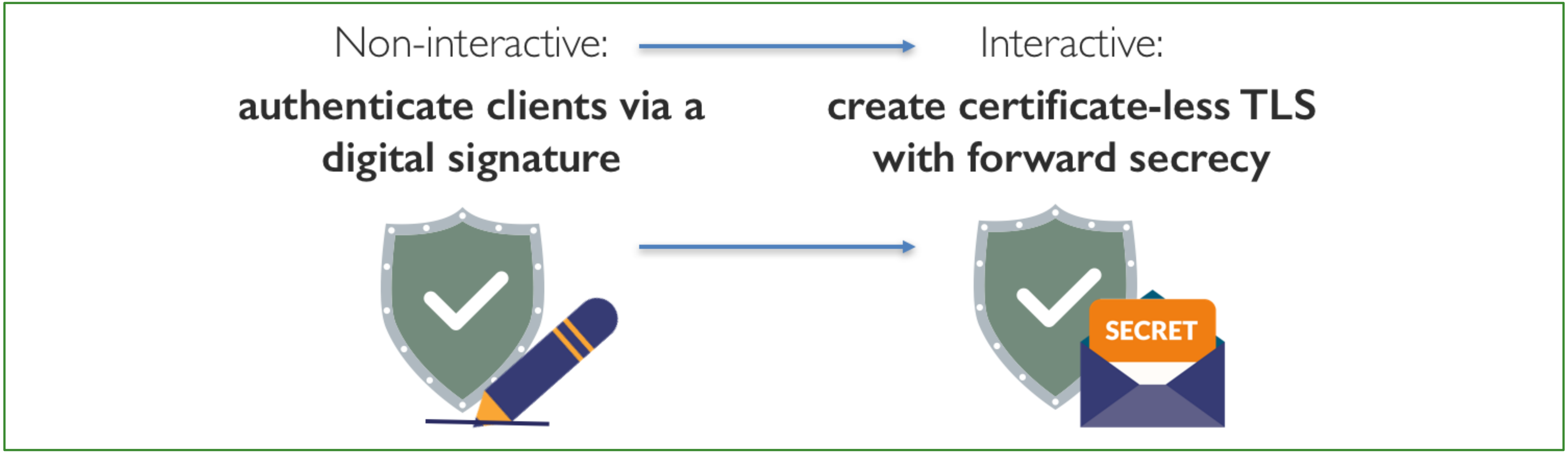
Improves
authentication / signature
user experience

Improves
authentication security
to multi-factor

Identity based cryptographic multi-factor authentication
and digital signature protocol that replaces passwords.
Milagro MFA runs entirely in software – it's browser / app friendly.

MILAGRO

Milagro TLS Library



Non-interactive: → Interactive:

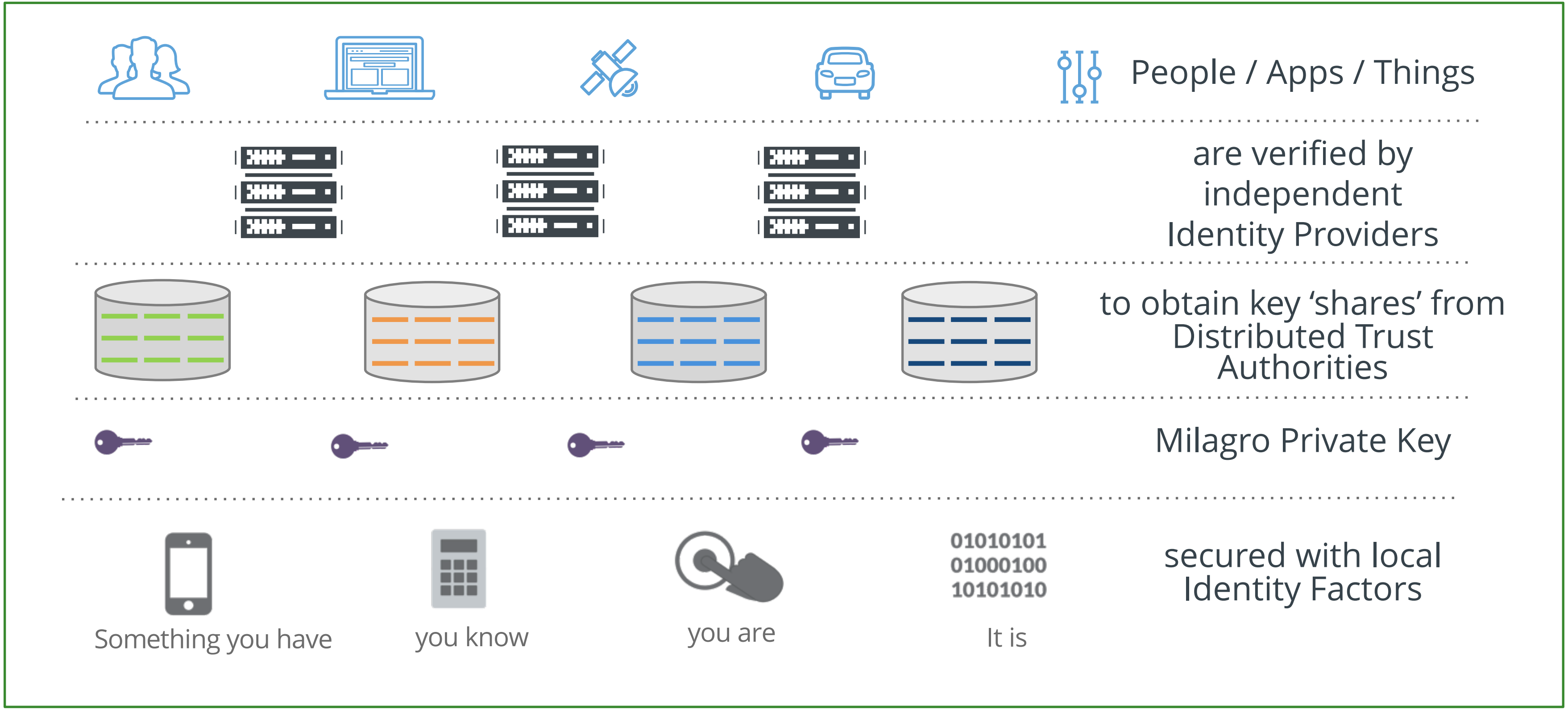
authenticate clients via a digital signature → **create certificate-less TLS with forward secrecy**

The same protocol run interactively creates an authenticated key agreement between client & server or peer to peer

~~PKI~~

MIRACL.COM

Apache Milagro is Built for Internet of Things



People / Apps / Things

are verified by independent Identity Providers

to obtain key ‘shares’ from Distributed Trust Authorities

Milagro Private Key

secured with local Identity Factors

Something you have

you know

you are

It is

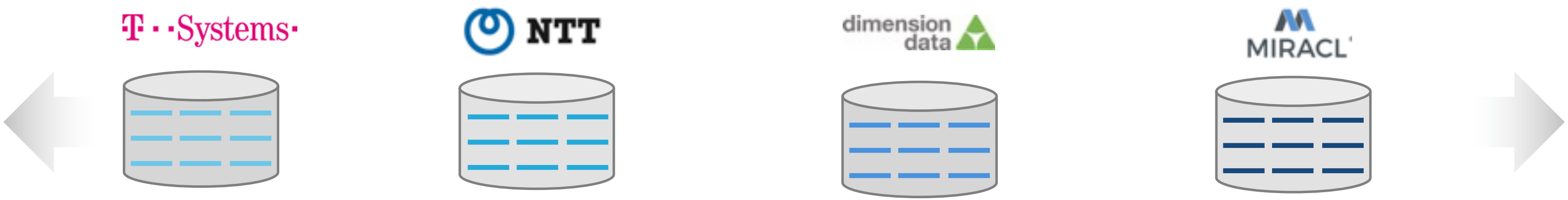
Milagro Ecosystem

PEOPLE / APP / THING REQUESTS SHARES OF KEYS FROM DISTRIBUTED TRUST AUTHORITIES

IDENTITY PROVIDERS vouch for the identity of people, apps, things to the distributed trust authorities

MIRACL BARCLAYS GOV.UK

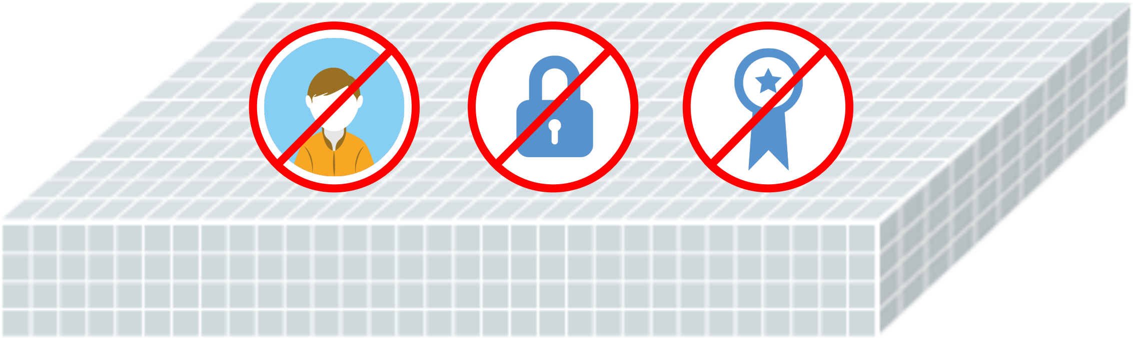
DISTRIBUTED TRUST AUTHORITIES issue shares of keys in the identity of and to a Person /App /Thing



DISTRIBUTED TRUST AUTHORITIES register proof of existence / create verifiable audit trails on BLOCKCHAIN

company	xx	xx-x-xx-x	xx-x-xx-x	xx-x-xx-x	xx-x-xx-x
company	xx		xx-x-xx-x	xx-x-xx-x	xx-x-xx-x
company	xx	xx-x-xx-x	xx-x-xx-x		xx-x-xx-x
company	xx			xx-x-xx-x	xx-x-xx-x

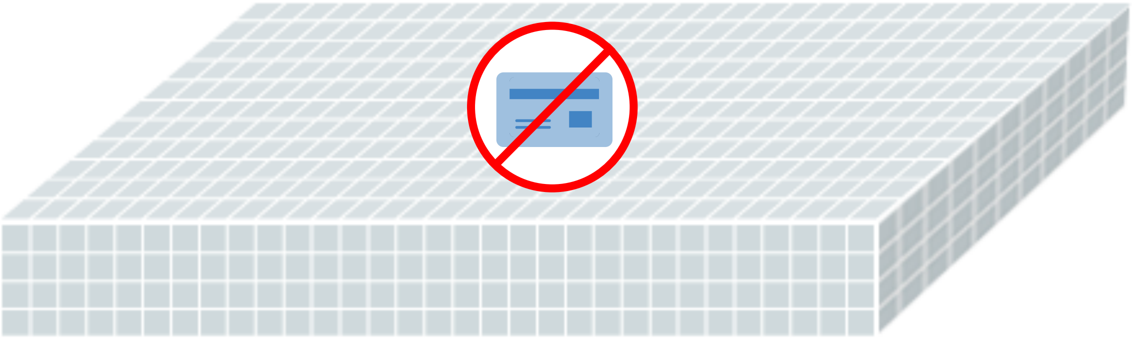
A Distributed Identity Based Cryptosystem for IoT and Blockchains



Blockchain Problem 1: Confidentiality and Transparency

Cryptocurrency transactions do not have enough privacy, nor the verification of identity, necessary to be compliant with banking regulatory requirements that deal with customer privacy and AML / KYC regulations.

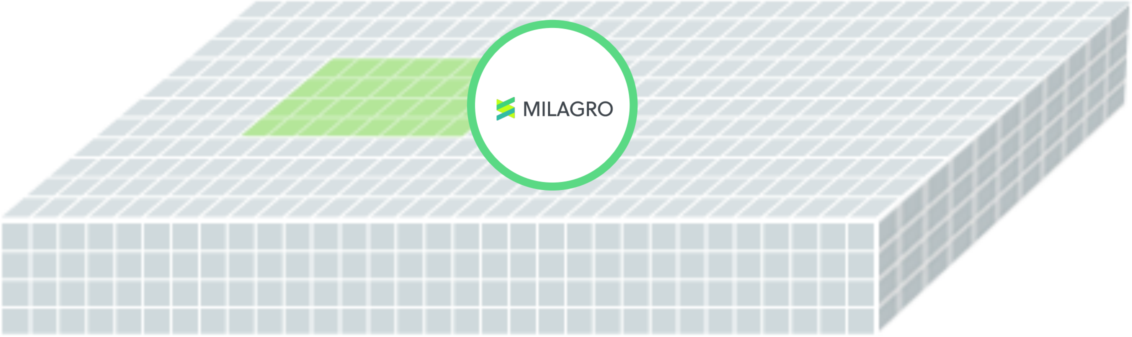
A Distributed Identity Based Cryptosystem for Blockchains



Blockchain Problem 2: Speed and Scalability

Cryptocurrency transactions by design can not be instantaneous. There is no way to create a capability to rival Visa's transaction network on Bitcoin's Blockchain (or any Proof of Work based Blockchain) without modifying the protocol itself.

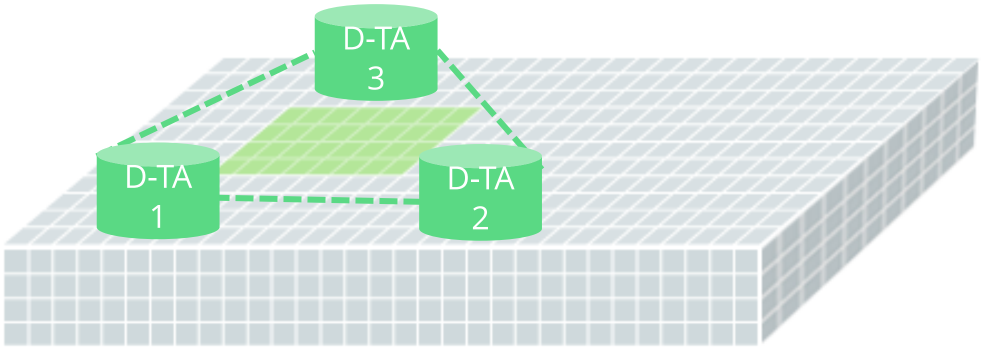
A Distributed Identity Based Cryptosystem for Blockchains



Blockchain Solution: Apache Milagro provides confidentiality, identity integrity and instantaneous transactions for P.O.W. based cryptosystems

Among transaction participants, identity integrity is assured.
Outside of the transaction, the transaction information is private.
Among all participants, the transaction is instantaneous, even if the underlying cryptocurrency is Proof of Work based.

Apache Milagro: How it Works



a) Distributed Trust Authorities are ‘anchored’ into the Blockchain and create a ‘partition’, which is a transaction ecosystem (merchant, vendor, individual, thing, etc.)

b) D-TA’s provide shares of ID based cryptographic keys (Milagro crypto tokens) to people, apps or things depending on ecosystem and use case within the partition.

c) Milagro tokens deliver identity integrity for participants within the partition (people, apps or things) and enable instant transactions within the partition.

d) Within the D-TA triangle, all transactions can meet KYC and AML requirements and are instant.

e) Outside of the D-TA triangle, all transactions are confidential and appear uniform.

Apache Milagro (incubating) Roadmap

Development of Enabling Protocols

- Milagro MFA mobile SDKs for iOS / Android (**completed**)
- Milagro 1-pass protocol for authentication and digital signature, released in Milagro MFA Mobile SDKs, Server and Javascript Client
- OpenID Connect Web SDKs for Milagro MFA federation

Development of Ecosystem (2017)

- Multi-Factor Authentication for Web and Mobile (**completed**)
- Distributed Trust Authorities (independent keys): Milagro D-TA code, Milagro blockchain client
- Identity Providers: Miladro IdP code, Milagro blockchain client
- Internet Of Things: IoT SDKs
- Blockchain: Milagro Proxy for blockchain and IoT

Goal - To quickly and collaboratively enable an independent security paradigm that provides strong authentication and cybersecurity across the web, over the Internet of Things, or on the Blockchain.

MILAGRO

milagro.incubator.apache.org