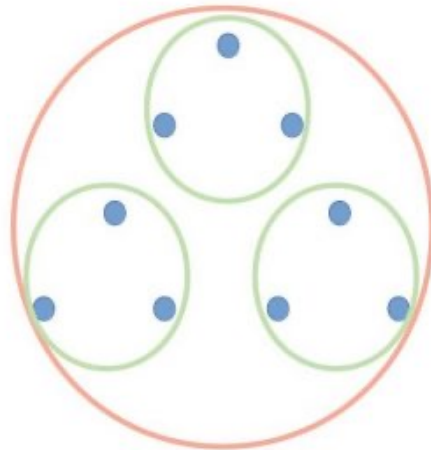


HymnParrotTalk

HymnParrotTalk: Hymn of the Underworld

Overview

ParrotTalk: Hymn of the Underworld. An always-on eventual promise replicating security framework, in Squeak & Java.



<ParrotTalk: Hymn of the Underworld>

The process of rotting is central to the functioning of an eventual promise system. This rotting is the process of shortening three-way promise scenarios, actualizing the Galaxy Object Scenario, such that intermediary connections are released in a controlled and deterministic way.

Derived from ELib's <<http://erights.org/elib/index.html>> Far Promises

Hoping to integrate with Croquet • <<https://croquet.io/docs/microverse/tutorial-Physics.html>> •

RC Promises / Alpha Whisper code

Grab Squeak :: <https://squeak.org>

I go into Tools->Preferences

- enable <Update from server at startup>
- enable <Open Tools Attached to Mouse Cursor>
- enable <Use colorful windows>
- enable <Extra debugger buttons>
- disable <Use Soft Drop Shadow>

to install HymnParrotTalk alpha, Dolt to...

```
Installer ss project: 'Cryptography'; install: 'ProCrypto.release.3'; install:  
'EParrotTalk.release.3'.
```

Crypto & HymnParrotTalk UnitTests

453 run in 0:00:00:29.305, 409 passes, 4 expected failures, 16 failures, 24 errors, 0 unexpected passes

Derived from ELib's <<http://erights.org/elib/index.html>> Far Promises

Hoping to integrate with Croquet • <<https://croquet.io/docs/microverse/tutorial-Physics.html>> •

Local Promises - Eventual Use and Examples

Then you can explore local promises and eventual sending

But...One of them there smaller mysteries appears...

Due to eventuality, a straight (self #assert: false) will not work, for some reason. Eventual Metaphysics.

```
big := (42 eventual * 10)
      then: [:x | x raisedTo: 10].
self assert: big isPromise.
assertion := big then: [:x | (big = 1708019812167782400000000000)
      ifTrue: [true]
      ifFalse: [self error: 'bad eventual result']].
assertion -> big.
```

The screenshot shows the Squeak Smalltalk IDE interface. On the left, a code editor displays the same code as the previous block. The middle pane shows a variable inspector for an ETuple object, with a tree view of its fields. The right pane shows a hierarchy browser for the class GozerStayPuff, displaying its inheritance structure and methods. A transcript window at the bottom left shows the output of the code execution, including the error message 'bad eventual result'.

Design Facets

- Promises/A+
- Eventual References
- Result Resolution
- Promise Pipelining
- Kitty :: Event-Loop Vat

Neighbor & Far Promises [BROKEN]

- Handled Promises
- Result Resolution
- Promise Pipelining

ASN1Type (Domain !! ASN1) Registrations & Designations <Tag|Name>

- Registrations are by ASN1Tag, in the Context Class <2r11>.
- Designations are by name, with ASN1Type, or DomainClass with ASN1Type protocol.

••• ASN1 HymnParrotTalk-v.3.9 •••

* encodings self-describing
* all msgs assigned numericTags & using a Context classTag of 2r11
* ASN1Tag has:
- classTag := 2r11000000
- is Constructed := 2r00000000
- numericTag: 2r00011111
* fullTag := (((classTag bitAnd: 3) << 6)
bitOr: ((isConstructed bitAnd: 1) << 5))
bitOr: (numericTag bitAnd: 2r111111).

••• ContextClassTag ASN1Types •••

Layer 5: Session v3.9

••• Errors •••

<0> • Duplicate
<1> • NotFound

••• Common •••

<2> • AdHocNamedStructure
<3> • RAW
<4> • Encoded
<5> • Encrypted
<6> • MAC
<7> • FEC

••• HymnParrotTalk-v.3.9 <8-18> •••

<8> • SubscribeKittyPKDH-v.3.9
<9> • ConnectKittyPKDHSig-v.3.9 [hashtag #SECURE_BOB](#)
<10> • GroupKittiesDH-v.3.9 [hashtag #SECURE_GROUP_ALICE](#)
<11> • Sig-v.3.9 [hashtag #SECURE_ALICE](#) [hashtag #SECURE_GROUP_BOB](#)
<12> • KittyLogin-v.3.9
<13> • KittyAuthority-v.3.9
<14> • PublishKittyPKCert-v.3.9
<15> • BridgesToKitty-v.3.9
<16> • KittySITREP-v.3.9
<17> • MeowMessage-v.3.9
<18> • ShutdownKittyInMsecs-v.3.9

••• Available <19-22> •••

<19> • Available-19
<20> • Available-20
<21> • Available-21
<22> • Available-22

Layer 6: Remote Promises

••• EventualMessages <23-26> •••

<23> • DeliverMessageOp
<24> • DeliverOnlyMessageOp
<25> • GCAnswerOp
<26> • GCExportOp

••• EventualDesc <27-31> •••

<27> • GiftPromiseDesc
<28> • FarDesc
<29> • RemotePromiseDesc
<30> • ImportDesc
<31> • IncomingDesc

ClassType

••• ClassType •••

To compile Spectres (Stubs from the Stub / Skeleton paradigm), with generated forwarding methods. And to build an ASN1Sequence Type for marshaling instances of this ClassType, as AdHoc passByCopy.

```
<
hashtag #ClassType
!! <ClassName
!! ASN1UTF8String
!! Exportable {true|false}>>
!! <IVarRecords
!! hashtag #SequenceOfIVarRecord
!! <IVarRecord
!! IVarName
!! ClassTypeName
!! ASN1Type>>
!! <MethodRecords
!! hashtag #SequenceOfMethodRecord
!! < hashtag #MethodRecord
!! MethodName
!! ReturnValueClassName
!! ReadOnly
!! RequiresTransaction>
>
>
```

Oasis, NeighborIntroducers & FarIntroducers

Franklin: Trinity Zisterz, Conductors, Orchestra & Choir

1. Trinity Zisterz :: Balancers of Power
2. Conductors :: Conductors
3. Orchestra :: Brokers
4. Choir :: Bridges

Hymn Full Stack

Thunks & ThunkStack

Hierarchy Browser: HymnParrotTalk_v_3_9

- ProtoObject
 - Object
 - ThunkRoot**
 - EncoderThunk
 - PTEncoderThunk
 - SSLEncoderThunk
 - Thunk
 - MessageDefragmenter
 - MessageFragmenter
 - SSHHalfDuplexProtocol
 - ThunkLayer
 - FrameBuffer
 - SSLReceivingFrameBuffer
 - GateKeeper
 - GhostbustersGateKeeper
 - HymnGateKeeper
 - InMemoryGateKeeper
 - ParrotTalkSessionOperations
 - HymnParrotTalk_v_3_9**
 - ParrotTalkSessionOperations_v3_6
 - ParrotTalkSessionOperations_v3_7
 - SSHTransportCap
 - SSLSessionOperations
 - GozerStayPuffMarshmelloMan
 - Layer5EndpointGozer
 - InMemoryGozer
 - ParrotTalkGozer
 - SocketGozer
 - SSLSocket
 - Layer6SessionGozer
 - FarGozer
 - ReplicaGozer
 - TrinityGozer
 - WhisperGozer
 - ParrotTransform
 - PumpedThunk
 - SocketThunk
 - SocketStreamEndpointBottom
 - SSHSocketEndpoint
 - UHFRadioThunk
 - SSLStageSecurityState
 - SSLThunkLayer

Selected Class: HymnParrotTalk_v_3_9

- all --
- control
- derived
- initialize-release
- protocol control
- protocol data
- protocol startup**
- think - accessing

Methods:

- processESqueakBridge:
- processESqueakContact:
- processHello:
- processHelloBridge:
- processHelloESqueakBroker:
- processResponse:
- processResponseBridge:
- processResponseESqueakBroker:
- processSignature:
- processSignatureBridge:
- processSignatureESqueakBroker:
- sendESqueakBridge
- sendESqueakContact
- sendHello
- sendHelloBridge
- sendHelloESqueakBroker
- sendResponseBridgeWithProtocol:encoder:dhPara
- sendResponseESqueaBrokerWithProtocol:encoder
- sendResponseWithProtocol:encoder:dhParam:
- sendSignature
- sendSignatureBridge
- sendSignatureESqueakBroker

Navigation: instance | class | ? | browse | senders | implementors | versions | inheritance | hierarchy | variables | source

Source Code:

```
messageSelectorAndArgumentNames
  "comment stating purpose of message"

  | temporary variable names |
  statements
```

Layer 8 :: MetaOasis Naming

<MetaOasis > BADZ-MV > Hymn > Whisper{HymnParrotTalk-v.3.9}>

Layer 7 :: BADZ-MV < Battlespace Analysis, Detection & Zelection MetaVerse Application >

<MetaOasis > BADZ-MV > Hymn > Whisper{HymnParrotTalk-v.3.9}>

<<https://croquet.io/docs/microverse/tutorial-Physics.html>>

Layer 6 :: Hymn Presentation

Hymn Metaverse Protocol, distributed secure capabilities.

<<https://lnkd.in/ejmBgHwG>>

••• Actors •••

TrinityZisterz ♀ ♀ ♀, the Cauldron and lots of Kitties

♀ ♀ ♀ + +

- Zisterz •>Conductors ••>Brokers •••>Bridges ••••>Kitty <Alice !! Bob !! Carol>

1. TrinityZisterz ♀ ♀ ♀

2. Cauldron

1. ••> Conductors - broad-cast Artifact/SyncTT

2. ••> Oasis

1. •••••> Oracle <Naming>

- ••••> active Publishers

- ••••> Artifactz <URL !! Guard !! TicketDispenser>

2. •••••> Ops <Tactical Operations Center>

- ••••> active Subscribers

- ••••> Artifactz <URL !! Guard !! TicketDispenser>

- ••••> <SyncTT> calibrated & broad-cast

3. ••> Brokers - re-broadcast SyncTT

- ••••> cache Artifactz in LocalOasis <Ops !! Oracle>

4. ••> Bridges - teatime stamping all traffic

3. Kitties

- ••> User/SystemSpaces

Galaxy Object Scenario

ELib's <<http://erights.org/elib/index.html>> Far Promises' Galaxy Object Scenario



The GalaxyObject scenario repaired.pdf 299KB
PDF

ELib's <<http://erights.org/elib/index.html>> Far Promises' Galaxy Object Scenario

Layer 5 :: Whisper Replicated Secure Session Design & Protocols

Whisper replicated encrypted sessions, over different Layer 4 NIC Transports, with teatime simulation clockin' in every network message, along with encrypted geo-location. Talkin' HymnParrotTalk-v.3.9

Architected with the following interacting componenets

1. **Gate <Server>**
2. **Multiple Gozers <Sessions>**
3. **GateKeeper <Operatrions running Protocols {state-machine & message management}>**
4. **KeyMaster <SecurityOps>**

ParrotTalk-v.3.6 (needs repairs)

1. IWant
2. IAM
3. GiveInfo
4. ReplyInfo
5. Go
6. GoToo

ParrotTalk-v.3.7 (needs repairs)

1. Hello
2. Response
3. Signature

HymnParrotTalk-v.3.9 (needs design, implementation & testing)

1. SubscribeKittyPKDH-v.3.9
 - {KittyID#/Sister|Broker|Bridge|Kitty/xGeoTT = nil/DH=nil?}
2. ConnectKittyPKDHSig-v.3.9
 - #SECURE_BOB
3. GroupKittiesDH-v.3.9
 - #SECURE_GROUP_ALICE
4. Sig-v.3.9
 - #SECURE_GROUP_BOB
 - #SECURE_ALICE
5. KittyLogin-v.3.9
 - {userID/passwd#}
6. KittyAuthority-v.3.9
 - {Caps/Certs}
7. PublishKittyPKCert-v.3.9
 - {<+|-> Sister|Broker|Bridge|Kitty/Cert}
8. BridgesToKitty-v.3.9
 - {set<BridgeURLKittyNonce>}
9. KittySITREP-v.3.9
 - {aliveness/alarm/cacheUpdate}
10. MeowMessage-v.3.9
 - {Rec#/Send#/MsgSym/priority/payload/xGeoTT ≈ Encrypted Geo & 12ByteTeaTime}
11. ShutdownInMSecs-v.3.9
 - {msecs}

SSH (needs repairs)

SSL-1.1 (needs repairs)

TLS-1.2 (needs repairs)

TLS-1.3 (future)

Signal (future)

MIL-STD-188-186 {UHF SATCOM} (future)

Layer 4 :: Multi-Channel Pumped Endpoint Transport

- **InMemoryPumpedEndPoint**
 - Software-Defined Shared-Memory Port-Map
- **SystemEndPoint**
 - SharedMemory
 - Pipe
- **SocketPumpedEndPoint {NIC}**
 - Ethernet
 - WiFi
 - Bluetooth
 - 3G/4G/5G+ Cellular
 - LoFi Transceiver
 - MIL-STD-188-186 UHF SATCOM Transceiver
 - Q-YarnBall ••• EntanglementTransceiver •••

Hymn Frame Specification

Anyways, FrameSpec is well specified. Able to extend, B! Signed & Sealed.

Ahh. 1 byte is 256 different AltSpecs. Yet, every message MUST have its priority, teatime, hash and size.

••• 14-byte AltSpec •••

1. First 8 bytes ::

- tags: 2 bits <MsgFrag !! AltSpec = True>
- priority: 2 bits
- teaTime (Nanos) <60 bits>

2. Last 6 bytes

- hash: 16 bits
- messageSize: 32 bits
- **[optional] FragCounter** the incremental message counter: 8 bits, for 256 messages.
- **[optional] SpecVersion**: 8 bits, for 256 Alts.
- ...

••• MsgFrag -> FragCounter •••

When **MsgFrag** is True, an extra byte provides for **FragCounter**, 256 sequenced messages.

When first fragment sent, **MsgFrag** bit set enters into **FragmentMode**. Set **FragCounter** to <1>, then subsequent fragments have incremental **FragCounter** set, where the **FragCounter** increments by 1 for each subsequent fragment.

At the end send 1 more message with **MsgFrag** cleared, to exit **FragmentMode**. Then a new Fragment sequence can begin anew.

••• AltSpec -> SpecVersion •••

When **AltSpec** is True, an extra byte specifies the altSpec, allowing 256 varieties. 0 is the base Frame Specification above.

We gotz the HOO HAA, the rest is the YAYA and the YAYA.

More Thunks and Modes

All these are easily deployed into the future across the network from the Sisters, adding their spices to the Cauldron. It's software-defined, evolutionary networking.

Modes

- ReplicationModes
- EncryptingOrders

Buffer Thunk

- downcall buffers outgoing traffic until flushed, by repeat-delay or upon explicit phase change
- upcall buffers accumulates traffic until condition (msgSize)

Multiplex thunk <3D Y, with 4 vertii>

ReplicationModes

For all, of 3 concrete sessions, one concrete session can be [hashtag #InMaintenance](#) while it gets closed and a new connection rendezvous occurs through the active Brokers.

Sessions can be closed from off a timer, to allow for crypto session cycling.

Sessions can also refresh their encryption by going through the Subscribe / Response / Sig DH cycle for crypto cipher cycling.

1. • **LeaderOneAtATimeMode** • Start with a Leader session and traffic until [hashtag #Closed](#) , then fail over to new Leader selection. This selection occurs by either the Client side (outgoing) or Connection side (incoming). Traffic is sent to only 1 session at a time.
2. • **QuorumOfTwoMode** • A Quorum out of potentially 3 sessions is 2. If the send is successful (no Closing or Transmit error) to 2 sessions, consider Quorum met.
3. • **FirstPassThePostMode** • Send traffic to all 3 sessions, if able. One transmission will arrive first.