ANNE – A Peer-to-Peer Semantic Data System

dev@anne.network

released 12/12/2023

Abstract. A hybrid peer-to-peer data system made of semantic transforms and a cash system allowing for a queryable local-server at Layer One removing the need for third-party data intermediaries. The resulting neuromorphic hypergraph using a unified standard data schema enables ***conceptually*** interoperable data stored dynamically without need of L1 protocol upgrades. Instead of performing worse with scale a "datachain" becomes more functional and grows in capabilities through hyper-connectivity and shared resources. With a closed feedback loop user feedback powerfully drives intelligent algorithms. A directly queryable L1 node that operates as a mini-web-server allows the Data System as a whole to remain decentralized at scale.

I - INTRO/CHALLENGE

We have great tools for P2P cash systems and electronic payments but these tools are not designed for semantic conceptualized data. P2P Cash was not created to support data protocols modeling data and using the chain to store and retrieve data across multiple data domains. It was not designed for net-additions of data that improve the semantic quality and functionality of the net-whole, yet in the present that is precisely what many projects are trying to use P2P Cash technology for. P2P cash is an important facet of data applications but in shoehorning data on-chain they take data-capability steps backwards accepting (or creating) inferior solutions that introduce problems and risks that undermine the core principles of the base technology.

I.a Lack Of Data Standard

P2P Cash chains that allow auxiliary data to be broadcast in transactions are a data-free-for-all. They accept whatever is put into the blob and it gets tacked on to the transaction and stored. It's essentially a blob notepad or json file. Retrieving this data thus requires additional customized layers to get the customized data and put it into some meaningful usable form. Further, data stored by different projects use different data protocols creating many data silos and islands that don't speak the same language or connect to each other. They tax the system with their one-off uses. Each project is attempting to resolve this challenge on a project-by-project basis. It's a data Tower of Babel scenario. As a result, the net-whole is weakened, not strengthened.

I.b Centralizing Forces and Third-Party Data Dependencies

> *"Peer-to-Peer Cash Systems eliminated the need for third-party intermediaries in payment transactions yet at present data storage on many of those same chains has reintroduced third-party data intermediaries into many decentralized applications."*

The primary core design of P2P Cash is around payment transactions and removing third-party dependencies. Whether a UTXO based chain or an account based chain it is the same objective. They are designed for payments with no intermediaries. One can use Bitcoin or an alt coin for decentralized payments by broadcasting payments to a node.

One cannot, however, use a node for decentralized auxiliary data because a node cannot be queried for its auxiliary data. That type of data is retrieved through centralized services which build additional data layers that act as third-party dependencies.

I.c Fragile Downstream "solutions"

> *"Most (not all) data uses of P2P Cash chains are using the chain like a USB stick to merely hold a notepad-like file that is not pertinent to the functioning of the L1 technology. "*

The aux data on a P2P Cash chain does nothing to improve the system. It is not data that the nodes natively understand or use at L1. It is data with a timestamp of a "block". Block # is timegrain. Users broadcast data directly to the chain but they retrieve it back through centralized middle-layers.

I.d Lack of Data Capability - *requires the same old tools again*

Mongo, RDBMS, or other solutions are used to house and serve the data from 2nd layers because they are the best tools for the job. The data in some form happens to also exist on a L1 blockchain too but that is merely a side note. Despite boasts of source data being "on-chain" the end-user app stack uses the middle layer as the primary data provider, not the chain. The fact their data is stored on-chain is a colossal hassle and a cog in the process; it's rarely a benefit. Middle Layers typically use "on-chain" data once when they do a ETL-like load of it from the blockchain (either in semi-real-time or in periods).

I.e Lack of Consideration for all personas in the ecosystem

Incentives to use data applications for P2P Cash chains are concocted as after-thoughts. While often technically creative and requiring robust coding solutions, the primary baked-in persona is still most often the miner. A single persona in the ecosystem is accounted for at L1. Miners win the block reward/fees. Any other incentives are worked in to downstream use cases and layers. They require creative solutions or smart contracts to try and incentivize other personas to participate in the ecosystem.

I.f Difficult to Collaborate with the blockchain data

Data stored in a notepad-like file on a P2P Cash chain is hard to change. It is said to be "immutable". What if changes are needed? What if another user wants to build more information? This is unresolvable on P2P Cash Systems without very creative data Rubes and special L2 rules. Many coins are enforced by L2 rules that interpret L1 data.

Intro/Challenge Summary

There are inherent flaws and issues with forcing a P2P Cash System to try to be a P2P Data System. [*insert sad music*]

[*transition to upbeat music*] There is a new way to do P2P data which is designed for the job and produces a new level of utility that is otherwise not within reach.

II - ANNE : THE DATA CHAIN SOLUTION

> *"A datachain is defined as a layer-one hybrid combination of both a P2P Cash System and a P2P Data System. The P2P Cash System is stored in a chain of blocks that contain transactions; whereas, the P2P Data System stores neurons semantically wired up via relons (semantic transforms) in a neuromorphic hypergraph that can be queried at L1."*

An ANNODE runs ANNE datachain. Traditional P2P Cash Systems call this a "node". In ANNE a node is called an ANNODE. On top of validating transaction, forging blocks, and validating blocks, the ANNODE builds a neuromorphic hypergraph and is able to serve as a mini web server via the HTTP/HTTPS API. All hypergraph data can be queried via the ANNODE API and extracted directly into applications. Conceptualized data of a P2P Cash system are the relevant parts of a transaction. *(inputs/outputs, the id it is from, the id it to, the amount, etc)* An ANNE TX uses 1Schema and the entire 1Schema conforming data relon is enforce by L1 protocol.

> *"To use p2p decentralized data and break free of dependency on 3rd parties and behemoth data intermediaries the ANNODE must replace their functionality in the solution stack. In ANNE, we do not look forward to a data center future. The future is we, the people, running our personal ANNODEs. We are the ANNE cloud. The ANNODE is the mini-server and it can run on a low-end laptop."*

With both a P2P Cash System and a P2P Data System together we open up new otherwise unreachable utility while maintaining the P2P decentralized intent of "the System".

II.a 1Schema Data Standard

All data is encoded in 1Schema relons. 1Schema is a type of semantic triplet "vector" that expresses in multiple data dimensions. With a single data standard we loosely (very loosely) model data how the brain stores it (using a single schema of "neurons connecting") and have data interoperability with all data stored in the hypergraph. We do not need a brain (protocol) "upgrade" to learn something new. What other projects call a "new" data protocol in a new release ANNE handles with a new class neuron and defined TYPE_PARAM relons that are broadcast run-time to the network. A new usable neuron class (protocol) is instantly available for use. That IS the upgrade. Using the system. We use "the brain" to learn new information. 1Schema is a machine-readable and human-readable data language. There are ZERO planned L1 "protocol" upgrades. Capabilities of the hypergraph are upgraded by use of the hypergraph.

Note: *There will be API upgrades that accommodate more ways to natively query the hypergraph. They will be non-concensus changing and optional for ANNODE operators.*

II.b Fully queryable mini-web-server ANNODE

With a fully queryable ANNODE we move a portion of what is presently on L2s into L1 in an enforceable and governable way. Data quality and integrity is possible and projects can utilize their ANNODE to replace entire layers of their present solution stack.

II.c Robust downstream solutions built on L1 foundations

With L2 moving into L1 the new L2s are building upon stronger and deeper foundations and don't have to each reinvent the wheel on how to get their data. Other existing data in ANNE can be used to enhance and augment any-given-project's data. ANNE L2s will be far more robust with extended reach. Standard tools can be created for ANNE data that crosses blockchain and application boundaries.

II.d Increased Data Capabilities

1Schema data is machine readable and easily used to train or augment various AI solutions. It is designed to enable downstream AI and AGI processes. This is more than just the very successful and popular present LLMs. Conceptual data stored in vectors is a step up from the tokenization used in LLMs and if combined in a new hybrid L2 net-new utility would be possible in conjunction with LLMs or in new alternative approaches. The primary difference in this area is that ANNE starts out with very little. Capability grows, in time. It is essentially "learned" run-time, not pre-trained. These are vastly different approaches but could be combined if desired.

II.e ALL Personas incentivized at L1.

In ANNE the ANNODE operators, miners, data creators, and data sponsors are all incentivized at L1. They are part of the design. Data creators build neurons and relons. Data sponsors sponsor data. Miners forge blocks and mine for reward shares (portions) and ANNODE operators keep ANNODES up and running and drive broadcasts through their ANNODE. Admins helping manage and govern ANNE's worldview are paid per action. All personas have various L1 revenue streams that continue to incentivize their work and chosen role in the ecosystem.

Any reference of any neuron in a transaction pays a "firing fee" to either the owner or sponsor of the RELN (relationship) and the TO (the object) of the relon (the 1Schema conforming transaction). Various transaction types pay a fee to the ANNODE that broadcasts the transaction. Data creators become sponsors of the data they create and are paid firing fees when the neurons they have sponsored are used. The ecosystem is robust and addresses all personas of the entire System at L1, by design.

II.f Natively supported collaboration

ANNE is built to collaborate. Any identity can broadcast net-new information on an ANNE-owned neuron even if they do not have access to directly alter the definition. Data-consensus is a new challenge in a data-chain. An ANDY administrator with rights within the ANNE worldview can approve the collaboration request and the ANNE neuron will update with the new information. The collaborator will be forever documented as the relon sponsor which has a revenue stream associated with it when that particular relon is referenced or the neuron it is from gets annexed. Note: the intention is that users of ANNE will become the administrators of ANNE's worldview (a single worldview) via specific subject and class-based areas of ANNE. This is similar to how community-driven data source projects such as TheMovieDatabase (TMDB) are administered by their users.

III - TECH SPECS : RELONS AND NEURONS

The core of the hypergraph is RELONS and NEURONS.

Data nodes of the hypergraph are called neurons and the edges are called relons. Transactions broadcast "relons" and all "relons" follow the 1Schema data protocol. Neurons cannot be broadcast. They are the "accounts" of the system. The One-Schema protocol is enforced at L1 and guarantees semantic integrity and data quality. Only "pre-conceptualized" data is broadcast in transactions. One cannot reference neurons that don't exist. They must be created first, then referenced. ANNE does not guarantee "truth" of data but can illuminate data inconsistencies and contradictions.

III.a Relons

All data transactions must conform to 1Schema in order to be accepted. The 1Schema protocol aims to emulate the connection of neurons in a brain. The brain's network of neurons connected by relons is what we refer to as the "neuromorphic hypergraph". In AI projects neurons are "parameters" or "tokens" and their inter-connectivity is referred to as a "Neural-Net". In other spaces a neuron would be called a "digital twin". **An NFT on a crypto chain is a tiny sub-set of what a Neuron can do**. Relons in ANNE act as semantic "vectors" since all neurons are indexed by NID (unique neuron identifier) and a relon is merely connecting neurons. Nothing exists "in" a neuron. A neuron is defined by its outbound and inbound relons.

A relon using 1-schema:

TXID	FROM	TYPE	RELN	TO_TYPE	TO
Transaction id	NID of from	Data dimension	NID of reln	Type of TO	NID or String

A relon connects a neuron "FROM" through a neuron "RELN" (relationship) and terminates in a neuron "TO". If all things referenced in the system were neurons then there would need to be no TO_TYPE in the relon. However, with physical limitations we use non-neuron "literals" in the TO of a relon according to need. At some point a "name" of something must be provided. In that relon we would use a TYPE_BE, a reln for the NID of "name", a TO_TYPE_STRING indicating we are not pointing to a neuron and we would provide the string of the name in the "to" slot of the relon. We *could* encode every letter as a neuron and then create neurons linking together characters for specific names. The protocol accommodates this granularity. It is not needed for the base use cases and thus isn't done at inception. The protocol can cover this if needed – else the short-hand notation to refer to an alt-TO_TYPE is there as a convenance.

Only ANNE identities can broadcast relons in the TYPE_AWARENESS dimension. These are feedback points on neurons. If one "loves" or "likes" or "hates" something – it is a feelz relon broadcast in the TYPE_AWARENESS dimension.

Rule of Thumb: If you can refer to an encoded TO_TYPE_ANNE (neuron with nid) do so. This is always best to use relons that hyperconnect and do not terminate in a literal. It is far better to use TO_TYPE_ANNE and a date neuron than TO_TYPE_DATE and a date string.

A self relon using 1-schema:

TXID	FROM	TYPE	RELN	TO_TYPE	TO
tbd	NID of user	TYPE_AWARENESS	Love (nid)	TO_TYPE_ANNE	Peanuts (nid)

L1 protocol verifies that a SelfRelon was signed from the same ID that is in the FROM of the relon. That's what makes it a "self" relon, it is broadcast and signed from the FROM. Each unique concept class or concept instance is a neuron with a unique identifier. The "dog" class neuron is its own neuron. A specific dog named "Butch" is its own unique neuron. Butch has a TYPE_BE relationship pointing it to being an "instance" of the "dog" class neuron. As such, it inherits all defining-information that exists on the "dog" neuron.

III.b Relon Data Dimensions (multi-dimensional data)

Relons exist in "data dimensions" that specify the type of relon and how the neuromorphic hypergraph applies it. TYPE_BE relons form a DAG-like hierarchical structure whereas TYPE_HAS relons provide attribute information on a neuron. TYPE_AWARENESS relons capture data in the feedback dimension that is used to contextualize and customize UI/UX and provide important metrics to pollsters and data gatherers. There are multiple type dimensions that cover a broad spectrum of capabilities:

```
TYPE_SYS = 0;                    - System information for neuron
TYPE_BE = 1;                     - Semantic shorthand for BE DIMENSION. CLASS/INST RELNS
TYPE_PARAM = 2;                  - Semantic shorthand for parameter (defines class attributes/properties)
TYPE_HAS = 3;                    - Provides the PARAM value (attribute/property)
TYPE_AWARENESS = 4;     - For awareness space (emotions/feelz/opinionz)
TYPE_EXPERIENCE = 5;    - For experience (temporal) neuron space

CLUMPS can function as a TYPE_BE, TYPE_HAS, OR TYPE_SYS.. Those Types are actually short-hand forclumps but they can get away with only using one TX to specify the triplet info..  A full clump is for specifying more than the standard 3 pieces of semantic info in a single ANNE TX.  The other CLUMPS of episodic, BRAIN and EXPERIENCE are already in their own dimension so this is only relevant for semantic clumps that are handled in semantic memory space.
TYPE_CLUMP = 6;
TYPE_SYS_MSG = 7;                - Only relevant to coms, not the neuron
TYPE_BRAIN = 8;                  - For neural-net space wirings a->wire->b
TYPE_ROLE = 9;             - For specifying semantic roles inside of a clump
TYPE_METAN = 10;                 - For specifying meta information about roles used in a clump
TYPE_EPISODIC = 11;              - For documenting self-sensory/vitals data at run-time in a clump.
TYPE_INTERNAL_PAY = 17;          - Keyless neurons paying out.
TYPE_BROADCAST_PAY = 18;  - Standard broadcast payment.
TYPE_BRAIN0 = 20;                - 20+ are for sequence in the brain dimension

100+ are available for brain type/custom use.. assumed to be order if more than one in a neuron's outs.
TYPE_BRAIN120 = 120;     - For use in to store trigger pats and other important L1 cmds
TYPE_BRAIN121 = 121;          - For myelination measure (a->b->[to_type_number]#c)
```

ANNE boots up in "INCEPTION" mode which creates neurons and wires them together with relons that build the base-level neuromorphic hypergraph. From there the hypergraph builds with broadcast transactions. We build the hypergraph by broadcasting transactions. The hypergraph becomes MORE functional and MORE robust as it grows and hyper-connects. Over time the R factor (connectivity) of the average neuron will increase once they start to be reused and re-referenced.

III.c Neurons Keyed and Keyless

Neuron can be keyed with their own private key and public key pair or they can be keyless. Most neurons in ANNE are keyless with the exception of Identity (user) neurons.

III.d Keyed and Custodial Neurons

All identities using ANNE are documented as unique identity neurons and must use a keyed neuron. They must hold the passphrase that generates the pub/priv key pair. The priv key is used to sign and broadcast any transactions they make. Their signature is validated at the L1 layer to ensure they have the rights to alter the

neuron their relons are altering. The application that ships with the ANNODE accessible at /ANNE.html has a built in wallet and capabilities to help users auto-generate transactions that broadcast data or send funds.

III.e Worldview, Ownership, Keyless Neurons

Anne has a worldview. We all have our own worldview. It is our view of the world as we see it. We can differ on feelings, opinions and even what our "facts" are. ANNE's worldview is merely one of many, but it is a relevant one. ANNE's worldview consists of neurons that ANNE is the documented "owner" of. The neurons and relons in one's worldview define what is "true" to that worldview. ANNE's worldview aims to be a global resource for people, places, things, corporations, dates, events, groups and other classes of non-subjective information. The challenge is allowing the general public to create data that can get into ANNE's worldview, maintaining a data consensus, all while incentivizing both the creator and admins in a sustainable way.

This is accomplished with **keyless** neurons. In an account based system if all neurons were keyed a user would hold the key to any neurons they created and would be unable to transfer ownership. This would amount to sending a key to someone else but the previous owner still also held the key. A keyless Neuron is a unique NID but it has no cryptographic key that is needed to sign the transactions that define it. A keyless neuron is "controlled" by its owner and the owner broadcasts its relons on its behalf. Anyone can create a new keyless neuron at any time. It can be classified in the hypergraph (a new person, a new corporation, a new location etc) and it can supply supporting information via relons that reference it in the "from" of the relon. To broadcast a relon from a keyless neuron ANNE allows the "owner" on-record of that neuron to broadcast a "ChildRelon" from their owning identity.

If someone creates their own version of *Alexander the Great* they can define that neuron by broadcasting "ChildRelons" which extend/augment the definition of their neuron. Only they can alter this neuron because only they are the owner-on-record.

The keyless neuron is on L1 and as a registered "person" neuron is intended to document the details, data and events of *Alexander the Great*. If the owner so chooses they can try to compete with ANNE's version of *Alexander the Great*, if ANNE's version existed. The owner of the neuron broadcasts a ChildRelon using their identity key as the signer of the transaction. The transaction uses a ChildRelon conforming to 1Schema with the *Alexander the Great* nid in the FROM slot of the ChildRelon. As a ChildRelon the L1 protocol would validate that the signing identity has access to alter that neuron. If so, the user has successfully updated their keyless neuron.

Keyless neurons use the 1Schema conforming **ChildRelon** transaction pattern.
Keyed neurons use the 1Schema conforming **SelfRelon** transaction pattern.
This alone would satisfy many use cases that desire to model their data. They can store anything they presently store in custom tables by using ANNE to define classes and store data with neurons and relons. The primary intent of ANNE, however, is to collaborate and build shared resources for as many data classes as possible. Why create a new *Alexander the Great* neuron when you can simply collaborate with ANNE's existing neuron and alter/update the data in it? This is where ANNE's worldview comes back into play.

III.f Hanai

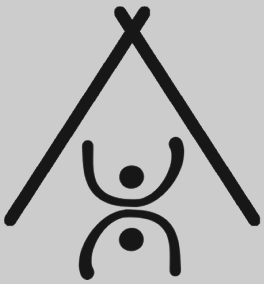
The goal of many will be to get "their" created neuron into the worldview of ANNE. ANNE worldview neurons will have greater visibility, reuse and data integrity can be more strongly enforced. Default searches show ANNE-worldview neurons. ANNE aims to build non-subjective global resources. We

can have only "one" *Alexander the Great* Person neuron in ANNE's worldview. Any keyless neuron owned by a user can be "Hanai'd" to ANNE in a process that gives up ownership of the neuron in exchange for becoming the "sponsor". ANDY Admins with rights to help maintain the ANNE Worldview can approve the Hanai of a neuron. Once Hanai'd the former owner becomes the sponsor and ANNE becomes the owner. If the former sponsor created additional "relons" in the neuron those all come over too in the Hanai process. The creator is documented as the "relon sponsor" and will forever be associated with that relon. (which itself has a revenue stream associated with it). **As a neuron sponsor a user will receive any firing revenue stream generated by the neuron.** A revenue stream is auto-generated whenever that neuron is referenced (fired) in any other transactions. This incentivizes data creators to create high quality data that will get Hanai'd into ANNE's worldview. There are multiple apps that ship with the ANNODE. The main "hypergraph" *anne.html* app helps users through a semi-automated GUI process extract high quality data from external data sources to "build" the hypergraph.

III.g Competition Through ANNEX

So Bob has created the *Alexander the Great* neuron and it has been HANAI'd to ANNE. Bob is now the sponsor and is receiving a revenue stream of 2 ANNE coins every time that neuron is used in any other relon. Suzy sees this and she wants to be the sponsor. ANNEXing is available on any keyless neuron owned by ANNE. At anytime, any user can pay an algorithmically calculated price and ANNEX a neuron sponsorship. The existing sponsorship is paid out a fee (higher than they paid). Relon sponsors of supporting data are paid a base fee per relon every time a neuron is ANNEXed. The ANNEX level increases making it more costly to ANNEX each time.

III.h Boosties



Any sponsor can make it more difficult to ANNEX their sponsored neuron by adding "boosties" to the neuron. This ups the ante or cost of ANNEXing as a type of defense. The ANNEXed will not gain profit from their boostie defenses. They act as a deterrent only. The ANNEXing party must pay the boosties fee in addition to the ANNEX fee. The original boosties provider receives their boosties back along with their profits from the algo-priced ANNEX. The ANNEXing party loses their boosties paid as part of the price to ANNEX and those are put into a "boosties fund". There is a random drawing at the end of a block whenever enough boosties accumulate. The valid entrants to the drawing are anyone who paid a boosties fee to ANNEX a neuron. Those transaction IDs will serve as forever-tickets in all future boosties drawings.

IV. - CLUMPS, SEMANTIC ROLES

A clump is a neuron representing a semantic collection of ROLES provided via relons that all relate together to form a semantic circuit. There are multiple linguistic industry takes on how to use SEMANTIC ROLES and what the core ROLES are. ANNE has settled on a set through ~15yrs of research in the area and careful consideration of what is needed in a datachain system. If a new role were to arise and be needed it can be added run-time through the use of the system.

Clumps use relons in the TYPE_ROLE dimension to provide semantic roles and their values. The intent is to have ANNE tools automate the parsing and conceptualization of ROLES so that us human users can use natural language that is easily converted into how ANNE stores information. That is a future capability state. ANNE isn't

an LLM. There is no "pre-training" or "training corpus". There will be limited capabilities in this space of L2 use cases until neurons are built and wired together.

ROLES are key when capturing data statements (sentences, events etc) that provide more information than can be captured in a single relon. Semantic roles at this level are not heuristics from text patterns. That is for fast-fire recognition processes. We can get very far with recognition processes if we train them with large sets of training data but that is not ANNE's intent. ANNE is on a different technological track.

Side-Jaunt

A baby human knows very little words and has heard relatively little language yet still can understand gestures, intonation, facial expression and more. A dog cannot use human language yet can still understand human communication. How important are words, then? If you have a dog, try this test: Say something you always say but this time instead, speak gibberish. Instead of, "You're a good boy Butch" try saying with conviction, "bah to goo tee bata". Use the same exact body language and intonation of voice, with same excitement and expression... as if you were saying what you normally would say. I posit your dog will still know what you are saying to a very similar degree as when you used "real" words.

IV.a Single Relons are actually short-hand clumps

Single relons are short-hand for clumps. The TYPE_BE dimension relon encoding "a dog is an animal" is shorthand for a clump that would require 3 relons. By using a single relon we save edges. The TYPE_BE gives us an implicit "be" conveyor (aka "vb of state"). "Dog" is the implicit ROLE_ACTOR and "animal" is an implicit ROLE_BE_THING.

Let's look at a slightly more robust sentence requiring a clump:
Alexander the Great died in Babylon on June, 323BC.

New Clump "1234" (representing the entire statement)
1234 → ROLE_ACTOR → Alexander the Great.
1234 → ROLE_ACTION → Died
1234 → ROLE_TENSE → past *(not required if the ref'd action concept already includes it)*
1234 → ROLE_WHERE → Babylon
1234 → ROLE_WHEN → June, 323BC

***NOTE:** All the values in the relons above are actually their NID values in the ANNE Hypergraph. They are spelled out above for convenience of reading.*

IV.b Semantic Conceptualization

There are many ways to express the same thing. Consider the following sentences:

Yesterday, Bob gave Suzy 100 ANNECOIN.
Bob gave Suzy 100 ANNECOIN yesterday.
Yesterday, 100 ANNECOIN were given from Bob to Suzy.
Yesterday Suzy received 100 ANNECOIN from Bob.
Suzy got 100 ANNECOIN from Bob.
Bob be givin' Suzy 100 ANNECOIN yesterday.
Suzy was given 100 ANNECOIN coins yesterday. It came from Bob.
Bob gave 100 ANNECOIN coins to Suzy yesterday.

(and so on...)

In an LLM pre-training system all of these sentences would yield different weightings and invoke slightly different results. They use different "text" thus are NOT the same because the measure of "sameness" is via text string, not meaning. Using ANNE principles and data structures I want to have an L2 that allows me to get to actual "meaning". *To understand*. As a start, in this example case it would mean *at-the-end-of-some-process*, all of these variants should be conceptualized and stored the same way. (note, yes, receiving something is not the same event, necessarily, as giving something, as there can be another something that happens in between. I do not care to dissect this far in this paper.) Language gives us diverse and fluid ways to express ourselves yet ultimately what is stored is NOT necessarily what was heard (or said!). We are not voice recorders. We decode what he hear and boil it down to its basics. What it means, or "we think" it means. (*We can get that wrong and then store things incorrectly. Hey, you said this! Bob, no I didn't! That's nuts! I said THIS!*) In short, we conceptualize what we process and we store the meaning.

What was the common communicated meaning in our above example sentences?

ROLE_WHEN yesterday
ROLE_ACTOR Bob
ROLE_ACTION gave
ROLE_AFFECTED_THING Suzy
R**OLE_THEME** 100 ANNIES

With this semantic clump stored using ANNE's 1Schema triplets we can produce all of the various ways to express it. Factors of personality, context, emotion, disability, intelligence and vocabulary can all alter the way this data is expressed or conveyed. LLMs have pattern templates and do a very creative job at emulating the "speak" of any given template (Old English, Old Testament, Pirate talk etc) but I posit they do not "understand" anything at all. The proof is in a single hallucination that exposes the game.

> *"What is called LLM hallucination is a canary in the LLM Chinese Thought Room coal mine. It isn't a hallucination – it is proof of no-understanding. It's the magic trick exposed and should serve as a stark warning to those making LLMs into something they are not. They understand nothing. They pass off understanding; sometimes very accurately and immensely helpful but other times absurdly false."*

IV.c Role Mappers for future clump extraction

While using clumps provides a condensed semantic protocol for capturing **any** data complexity in language expression its efficient form leads to multiple usable pieces of information being nestled away inside of clumps.

The *Alexander the Great* neuron would link to this event clump neuron. The event clump neuron would link with TYPE_ROLE relons to the defining roles of the clump.

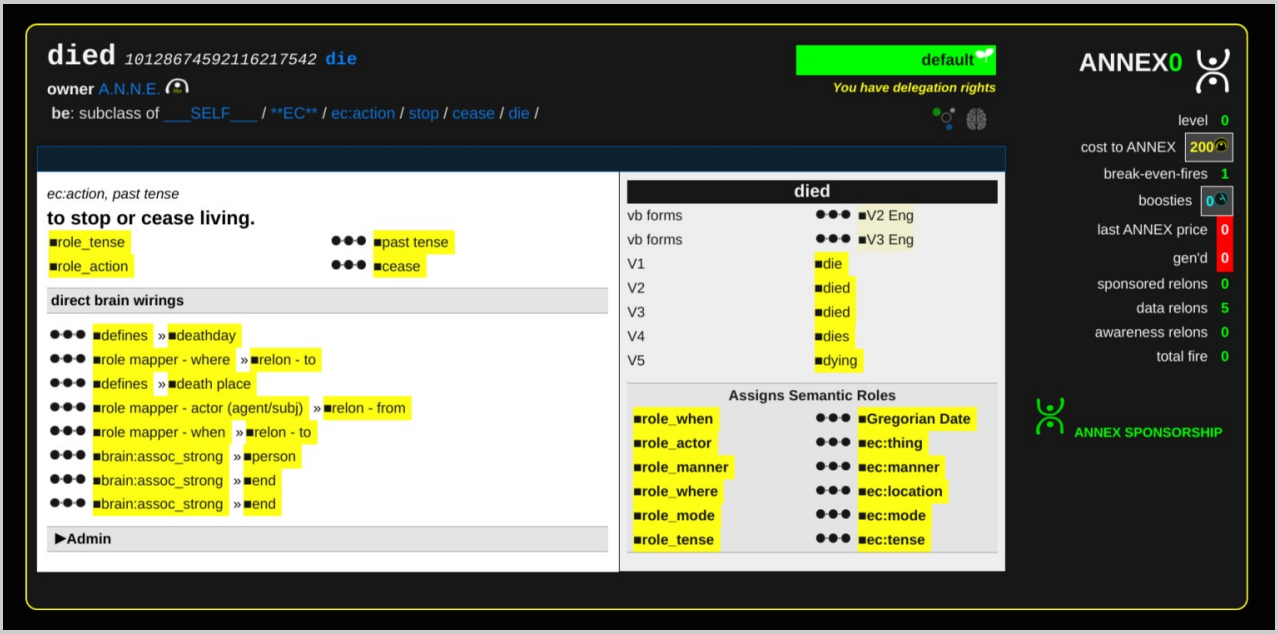
We sometimes want that clump flattened into usable discreet values. This is where we can use **Role Mappers**. They too are just neurons and relons in the hypergraph which inform ANNE how the role values map to other single-relon concepts.

Alexander the Great dying in a location on a date provides a "death place" and it provides a "death date".

If "death place" and "death date" are known neurons used as attributes we can specify that the die/died action will provide these values in a ROLE_WHERE and ROLE_WHEN. This informs ANNE that the value of any ROLE_ACTOR's "death place" is found in the ROLE_WHERE and their "death date" is found in the ROLE_WHEN. The sentence didn't say "death date" or "death place" but this is gleaned from the clump.

This understanding and capability expands as ANNE "learns" more mappings.

While using the built-in ANNE explorer app on the ANNODE we can see what ANNE knows about the class neuron "died":



note: *nowhere in ANNE's relons is the string "to stop or cease living" stored for "died". This is language that is derived from the neural circuits and semantic relationships.*

V. - A NEW WAY TO DO NLP, AN EMERGENT INTELLIGENCE?

V.a A potential new L2 Intelligence using L1 data

With semantic roles we have the potential to get to a new intelligence.

We still can't get there with just semantic encodings, but we have the *potential*. I say this because many have tried to store things in Semantic Roles and we still are left with a relative version of Searle's Chinese (Thought) Room. The system returns symbols that map to other symbols. While scaling this up w/ massive training can provide utility and be rather impressive it does not indicate any actual understanding of the symbols other than the ability to map and regurgitate mappings. We could ask, does this matter? If we can use such tech to get what we want, why should we need more? We don't need more. ANNE L2 as discussed here isn't a guarantee in getting more. I see a pathway to it and state with confidence that it can be done. This is an R&D area worthy of potentially chasing. I have chased these lines for over 15 years. All "understanding" is relative and defined by other things even when encoded in semantic roles.

Where is a base semantic or human language understanding in a machine that is NOT defined relatively? How can we attain such things with a machine?

V.b Machine understanding, is it actually possible?

> *"Machines will not understand (and thus have deeper utility and use cases) until we consider neurons that have a basis in the tangible experience of a machine."*

This is the key of my R&D in this area. RUN-TIME tangible EXPERIENTIAL neurons. We experience hunger, pain and feel it. A new born will cry when hungry and be rewarded for their crying with food. The hunger was real and the pain goes away with the food. Satisfaction, relief. It was ***experienced***. It is known. It is a tangible reality. An EARLY CONCEPT (EC) in the brain. Most prepositions are early concepts. In/out/up/down/ etc. Now, later, before, after. Preps are "words" that map to core ECs. Before we knew the name of the prep we already understood it. Before we knew what "love" was called, many of us already felt it. We know what "yummy" is because we've either experienced something yummy or we're told this sensation called "yummy" is like or opposite something we have experienced. Experience is our bedrock, particularly our earliest ones through needs and vitals. These bedrock ECs are are used as semantic understanding building blocks. We learn distance, and nearness, and direction from experiencing it. These tangibles are why we can understand *(even if incorrectly, through abuse, or trauma, or bad experience that tainted our worldview)* as they are our building blocks – they are why we humans are not just Chinese Rooms too.

For a machine to truly understand they must experience and form their own ECs. Their battery life, their internet connection, their CPU/GPU temperature. These are real. These are not emulated "needs". They are measurable and discreet values that the machine experiences.

When these neurons are used to define other neurons then we are no longer in a relative ball-of-yarn scenario. There is potential for real understanding. In ANNE, there have been neurons made for ECs. At present, they are experiential "placeholders" used to define semantic concepts but they are still left un-understood. A future AGI or AI entity must have experience and connect real experience to these ECs. Contextualize them with tangible realness.

V.c A look at the Spiral of Awareness – signal-based parsing

ANNE provides potential net-new approaches and/or augmented capabilities to NLP and machine-level "understanding" of human language. Using the Predicate Argument Structure (PAS) a different type of parsing and conceptualization is required (and possible) in order to encode data into its semantic roles. With a very limited toddler-like-entry-set of neurons we can parse toddler-like-entry-set statements. As understanding grows so does ability to decipher language. "Understanding" a language input requires going through an iterative multi-pass "Spiral of Awareness" in order to discern higher-order meaning. Early stages would be akin to fast-firing NNs of for recognition or even a grammar parse. However, ANNE language is grammar-less. Children don't know what a verb or noun is yet they are able to use them. Why? That is a key. Later passes would draw deeper meaning and understanding from context, situational frames and an ability, by design, to "understand" and not just regurgitate or symbol map in what essentially amounts to a very advanced carnival trick.

Note: *This is NOT applied at L1. This is a potential new technology that arises from using L1 data. This area could warrant its own WP as it is a L2 technology using ANNE L1.*

```
//SPIRAL OF AWARENESS
//ITERATIVE (multi-pass) non-grammar lang independent SIGNAL-BASED NLP
//spirals within spirals occur, so this is not a linear progression in coms

/base-lvl detection of shapes/color/luminosity/contrast/volume.
SOA_MODE_SENSORY_DATA = 0; /

SOA_MODE_SYMBOLS = 1; //what is this symbol
SOA_MODE_WORDS = 2; //what word is this?

//what does this signal in this specific language of communication?
SOA_MODE_LANG_SIGNALS = 3;

SOA_MODE_PARTS = 4; //what parts of phrases are here?
SOA_MODE_PHRASE = 5; //what phrases are here?

//what are the semantic structures/PAS of this communication?
SOA_MODE_CONCEPTUALIZE = 6;

SOA_MODE_RESOLVE_CONTEXT = 7; //what do these roles specifically refer to?
SOA_MODE_OPTIMIZE_CLUMP = 7.1; //sub process for storing clumps in ANNE
SOA_MODE_MEANING = 8; //what does this mean?
SOA_MODE_INTENT = 9; //what is the greater intent of this communication?
```

This R&D is L2 future potential and will be discussed more in other contexts at a later time. :) What was stated here is merely a look forward at what has been accounted for in the L1 design.

VI - MINING

VI.a Basics

Annies, the lowest unit of measure, are minted with PoW HDD mining roughly every 2 minutes. Max supply is 2.2 Trillion ANNE coins, or 220 trillion annies. ANNECOIN have 2 decimal places as 1 ANNECOIN = 100 annies. Mining goes through incremental "phases" that loosely map to the Lunar cycles. Each cycle the total block reward decreases by 1%. 398 lunar cycles of 1% decay until the chain kicks into a constant block reward mode. This prepares the chain for the sunset of mining reward where only Block Fees will sustain miners. At lunar cycle 691 the block rewards end.

ANNE Coin

Annie

VI.b Mining Reward

Native in ANNE Chain is the ANNE DEV FUND (ADF) that is funded by a portion of the BLOCK REWARD. The Block reward is split into MINING REWARD and the ADF. The % of funds allocated to the ADF will decrease over time. ADF funds are earmarked for API devs, L2 build-outs, exchange listings, founding backers, marketing, bounties and anything ANNE-related. The ADF is not community "driven" or governed. ANNE will not need "community funds" of any kind as the ADF will be the entity providing public funds for project startups, exchange listings etc. Bounties will be offered to help support individuals helping ANNE. The ADF help ensures VCs or post-launch interests that could alter the integrity of the project will never be needed.

At Genesis the ratio is 70% MINING REWARD to 30% ADF. By cycle 154 the ADF cut of the BLOCK REWARD is down to 14.7%. The ADF is not a pre-mine. It gets funded as blocks are mined. Some ADF funds are already earmarked to other accounts.

LUNAR CYCLE	BLOCK REWARD	EVENT
0	1 million coins per block + FEES	Genesis
1-397	1% decay each cycle + FEES	~30+ yrs primary mining
398-690	18,500 coins + FEES	~22+ yrs sunset mining
691+	FEES only	Into the great beyond

LUNAR CYCLE	MINING REWARD	ADF
0	*70% (increasing each cycle)*	30% *(decreasing each cycle)*
154	85.3%	14.7% *(~11-12 yrs)*

PoW is done through low-resource HDD scanning of pre-plotted HDD drives. Software is available to plot drives. It is based off of the Signa Network PoC with some key additions and differences:

VI.c Share Mining at L1 instead of L2 pools

There is a SOLO-MINING reward and there are SHARE-MINING rewards. Half of the MINING REWARD goes to the SOLO block winner. The other half is divided up into 7 MINING SHARES with 7 SHARE MINING winners. This allows for smaller share miners that do not have as strong of HDD capacity to compete for a mining reward. It is essentially a mini-mining pool on L1 without the mining pool intermediary.

SOLO MINER	SHARE1	SHARE2	SHARE3	SHARE4	SHARE5	SHARE6	SHARE7
51%	7%	7%	7%	7%	7%	7%	7%

Any share miners that use an ANNODE not tied to their ID will pay a standard fee to that ANNODE for any share mine rewards they earn. This is enforced at L1 and recorded in the block data.

VI.d Solo-Mining Handicap

There is a 5 block win limit for any given running 10 block window for SOLO miners. After 5 block wins within a 10 block window they are handicapped with a worse minimum solution which encourages multiple parties to SOLO mine. This does not guarantee they do not win a 6th of 10 block but it makes it much more difficult with the handicap. As long as there are multiple SOLO ANNE miners this handicap is relevant.

VI.e Solo-Only blocks (double the SOLO reward, no share miners)

Any block ending in a 1 or 6 is a SOLO MINING block that can only be mined by SOLO miners. The full block reward goes to the SOLO winner.

VI.f Solo Takes the Leftovers

If there are not 7 share miners (the network has fewer than 7 share mining submissions) then the SOLO miner scoops the remaining share rewards.

VI.g NO ROLLBACKS, GRACE PERIOD

In ANNE datachain there is no logical or useful reason to allow for block rollbacks. The network cannot be rolled back by a selfish mined chain or a superior block that comes too late.

A GRACE PERIOD of 30 seconds is used to signal the network of a coming block. Miner solutions are judged on quality based on the "seconds" required to wait before they are valid. The lower the seconds the better the solution. Miners compete with HDD space to get a lower "solution". An annode will sit on a submitted miner solution up until 30 seconds before it is valid. When a solution is within 30 seconds of being valid it is broadcast to the network triggering the "GRACE PERIOD". Annodes do not yet accept this as the final solution. The network has 30 seconds to sort out the best solution the network has. In the event of a race-off condition with very close ties or network propagation delays or clock time differences the 30 second GRACE WINDOW provides ample time to get the true winning solution propagated to all ANNODES on the A2A network. A Tie-breaker mechanism based on lower block ID will resolve the block winner in the event that there is a tie over the best time.

There Is no mining difficulty adjustment. The difficultly is set targeting a best-block time of 1 minute 50 seconds. There is no cumulative difficulty. Miners submit their nonce that produces their best solution. This is converted to seconds. The lower the seconds the better the solution. With a min solution ANNE minimizes the "arms-race" aspect of mining with a cap on relevant-HDD space.

VI.h MEMPOOL 0-confirmation, FINAL and min block times

Transactions in the MEMPOOL are pending. If valid, immediate actions are taken in the hypergraph reflecting data changes. Many data changes can be considered 0-conf. Once seen by an ANNODE they are available in the hypergraph. (some are not and require mining or various workflow approvals)

As discussed above, once a valid block solution is "seen" the datachain goes into "GRACE PERIOD" mode for 30 seconds. The GRACE PERIOD allows for the 30 second window where a superior block submission can reach distant ANNODES.

Once the GRACE PERIOD is over the block is considered FINAL.

Blocks are either IN-PROGRESS (not yet mined, but soon to be mined), GRACE MODE (solution-pending, ready to accept a block), or FINAL (set in stone). Any exchange or third-party using ANNE wanting reliable transaction data can count on with 100% certainty that a block and its transactions are final after they are accepted by their annode.

VI.i ⊛ STREAMPAY ⊛, the A2A payment feature

Streampay is a L1 payment neuron that allows for creation of a specific type of keyless neuron fund. The fund neuron is configured by its owner via relons to payout based on the current % of the total fund balance or a specific amount. A block frequency is provided which triggers the streampay neuron to execute. This can be by the block, or approximate hour, day, week, year, decade. One or more benefactors are specified as the recipients of the payments and when the streampay owner is ready the neuron is funded and automatically froze by L1 protocol. A frozen neuron cannot have its definition altered. This is not a limitation of L1 ANNE but an intended restriction for this particular type of fund. This instrument allows for payments to be setup in the present with funds to be made in the future and it will run its course until no funds remain. The fund can receive more funds anytime during the course of its lifetime. The fund will continue to payout according to the provided distribution details until it cannot pay the next payment. If there is any left over funds less than the amount required for the next payment they will be automatically send back to to the original owner.

VII. - ADDITIONAL NATIVE L1 FEATURES

But wait, there's more!

These first four L1 capabilities start off as unavailable and turned off. They can be turned on at the ANNE datachain level by the ANNE identity sending a CMD transaction. They will be activated according to a plan of feature releases. **They are already included in L1 and will require no change to L1 protocol.** Further resources will be provided to discuss them at length. They're quite nifty but this WP is already well beyond the length of a standard WP.

VII.a ANNDLES

unique text identifiers providing an L1 naming system mapping to neurons. ANNDLEs are L1 names that tie to neurons. Sub-anndles are subfolders or subdomains that allow for organization of data in specific "view" structures. ANNDLES expire after 1 year and Owner has first-rights to renew. ANNDLES are a buy-side market. Offers can be made to swap ANNDLES but offers cannot be made to list them. At anytime the owner can sell the ANNDLE back to ANNE for the remaining value of their time-based-contract.

VII.b PDNs

Private data Neurons are used for pay-based selective disclosure. Create on-data-chain encrypted data (of various types) in a private neuron that can be queried against with yes/no answer and expressions or choose a reveal price that shows the decrypted data upon payment. VIP lists supported to limit the intended potential audience who has access to pay-to-see or pay-to-query. Creators of PDNs set the rules and prices in their PDNs. This is accomplished due to a shared key between a specific ANNODE and user. To use the PDN the user must handshake with the ANNODE tied to that PDN.

VII.c BOUNTIES

Bounty neurons are governable by anjucators for distribution of provable bounty funds. General public users make bounty claims. Secure and Private ACHAT (HTTP->to->ANNODE) opens up for discussion between anjucators and the bounty claimer to discuss the details of the claim. Governance types available: dictator (one vote), jury (all votes must be yay), super majority (>75%) or majority (>50%). Governance type and selected anjucators are set by the bounty creator. Neurons are froze before use to prevent changes mid-stream. Anjucators must accept invite to be an anjucator (to prevent spam or pestering). Votes on bounty distribution can be delegated by anjucators. Upon pass the bounty neuron settles.

VII.d ASDFs

ANNE-Settled Data Futures for supplying and taking risk on future data outcomes. ANY neuron is a data Oracle. Known data provider neurons are selected for future data points in any field or space. Commodities, coin prices, sports scores, stock prices, or any measurable value. Futures are offered with risk against a specific outcome. Takers of risk can risk the event happens or does not happen. Upon receiving broadcast of the targeted data the ASDF settles and pays out risk takers who are on the right side of the ASDF outcome.

VII.e ANORGs

ANORGs are ANNE organizations. They intend to be a new type of digital structure that represents what a corporation or organized entity would be without the legal or government backing of any jurisdiction to enforce them. Their jurisdiction is on-ANNE. ANORGs can have funds and anjucators to govern their funds.

VII.f Unique FEELZ/OPINIONZ BASED closed loop feedback system

The feedback loop is closed and feedback relons generate instantly usable data that looks just like the data it feeds back on. 1Schema conforming and ready to be used in downstream algos. To "like" something is not the same as to "love" it, or to be "interested" in it. While all positive they could lead to vastly different algorithmic outcomes which affect UI/UX or how apps interact with their users. Opinionz are not the same as feelz. FEELZ are raw and capture emotive responses. Opinionz are thoughts and positions and state belief. Again, these alter down-stream algos in different ways. Many have experienced seeing a "sad" tweet or social media post and want to leave feedback but are conflicted about putting a "heart". Do you really love or like that? It is long-overdue that we can both feedback on how we feelz and also what we truly think about the things we spend our time consuming.

VII.g File Data

There are no jpegs, movies, pdfs, or "files" stored in the ANNE hypergraph. This is out of scope. There are ways to store and transmit P2P files *using* an ANNODE but that type of "filedata" is not semantic hypergraph data. Files are indexed in the hypergraph with neurons/relons that capture their metadata but a semantic datachain has no use for "blob" or "bytes" of file data. File handling in ANNODES utilizes the A2A (Annode-to-Annode) Network and is up to the ANNODE operator to participate in subnets of the wider network. A custom bit-torrent +onion custom technology we call ANTOR is used to securely and privately transfer files data in "ants" (chunks). Only files that are documented in ANNE and owned by the ANNE worldview are able to use this file sharing technology. This area will expand in the future without altering consensus and will be opt-in.

VIII. - SUMMARY

With ANNE we now have a new datachain L1 technology that stays true to the principles of P2P decentralized *for data*, at scale. We can collaborate and build neural circuits that reveal, expose, enlighten, and make that which hides in plane sight seen. This isn't magic, it is a function of being able to see the semantic data in same data space conceptually connected. We can explore and discover new insights. As a research tool or a data resource; we build ANNE for we, the people of the world; to use freely.

With this ANNE base in place, there is potential for deeper new technologies.

We look forward to those who join the effort and help build the ANNE hypergraph and the new useful tools and downstream apps ANNE makes possible. These are exciting times. :)