

📄/hyperpost/~gyuri/do/🌱/🌱/📄/index.html

📄hyperpost-gyuri-do-🌱-🌱-📄.html

protocol: 📄

- At the top 1 can read the full Universal Resource Name of this document
- It is a permanent name interpreted as follows
- 📄: utf8Icon that identifies the protocol/infrastructure that is used to create/access the resource
 - True URNs do have the ability to be amenable to the pun to be interpreted operationally to actually obtain the uniquely named resource in its current version
 - and in fact have access to its full attributed version history
 - This is why we call the IndyWeb designed to be *Integral MetaReflective, Homiconic, Meta-Designed*
 - *Integral because*
 - not separating out common required capabilities into separate services or layers
 - but insist that versioning of resources is integral to the very concept of all infrastructural alternatives that can be used
 - *MetaReflective*
 - all information/capabilities are presented embedded in an associative Complex that capture in its neighbourhood all the information needed to comprehend informatly what needs to be understood fo its use and functioning
 - Homo-iconic
 - Everything is formulated at the start as a Rich HyperText document System
 - with intellectually transparent notation
 - that itself is self-describing, metareflective and homoiconic
 - so that working instantiation of systems can be developed that make it all operate
 - So it is not the old way of write a specification and then implement it
 - but co-evolve the intential affectual formulation of the system in its own terms
 - so that it can actually be experienced and interacted with to validate
 - that the system delivers the experience, results it is described to do
 - Since it is extensible, forward compatiibty can be assured
 - New versions has to be developed in a way that they would work with previous version
 - upgrade on the fly if need be
 - Each set of information tht is created is shipped with capabilities that are needed to work with them
 - this is because information is formulated descibed in explicitly articulated, formulated and operationalized intentional terms
 - and implemented capabilities that allow the morphic transformations affordances to operate as described and intended
 -

player: /hyperpost/~gyuri

- global unique name of a Peergos Account
 - although this had been created with Peergos.net service
 - but the account identity is not tied to the name of the service
 - because one can always run Peergos locally
 - and login in with any Peergos account Name
- /~: Use the tilde as a subject marker, in this case a nicknamed human subject created by whoever is the owner of the hyperpost peergos account
- /gyuri: that's me (Gyuri Lajos, the author of this document, the
 - Lead EnvisionEer of the IndyVerse/Web/Net and the one that dreaming up in Intentional Universal HyperDocument and intentionaly explicitl elaboration the system starting with HyperPost

intent: do/🌱/🌱/📄/index.html

- **// intentional experiential experimental incremental exploratory wholistic constructive system, infrastructures and capability design**
 - Open Source, Open Constructs, Open Sauce
 - design for radical altruism, create true value for the long term, the Long View
 - choose protocols and systems that support these values
 - not for short term extractive personal gain
 - make it unstoppable, un-enclose-able, un-surveilable, autonom, private first
 - local-first, offline-first, permanent, future compatibel, evergreen
 - coevolutionary
 - with full provenance and mutual audit trails
 - not just decentralized but People Centered
 - commons based peer produced constellations that are exchangeable, extendible tinkerbabe
 - is what is needed, and it leads to
 - articulation/formulation/characterization of information in situ, and always
 - with an explicit intent, an attempt to explicate and externalized in
 - structured explicitly named elements in a concreate situation
 - instance first,
 - Infrastructure first.
 - instead of starting with nonfunctional requirements and design the system based on those choices
 - think hard of the value choices, come up with infrastructure vision that satisfy these values and goals and envision what they mean for all scales
 - The needs of individual human beings and their innate need of co-munication-laboration at the center, need for autonomy, long view lifelong availability,
 - control their data to control their destiny
 - and envisaged their operations at all required scales and constellations
 - make your design omni optional, do not make infrastructural decision
 - independently of the need sof the whole
 -
 - there is no separation of information viewed in the abstract devoid of use and intent
 - as is customary,
 - reject the very concept of abstract data structures and information model
 - all information arises in some intentional contexts, or rather *complex*
 - Make it up as we go along
 - not trying to design in the abstract with some vacuous doomed notion of permanence
 - but rather face up to the fact that information structures of necessity arise and co-evolve through use
 - instead of trying o pin down something that would stand the test of ime
 - understand that any conceptualization 1 makes will of necessity be needed to be changed as the systems evolve
 - and focus on making such changes easy and cheap to do regardless of how
 - radial they may be
 - It is a way of saying, that we must reject the pancake philosophy of LISP that
 - in a sense rightly said,
 - make your system work with your minimal,
 - initial understanding and be prepared to practice the pancake philosophy, which says the first one you make should be thrown away
 - and start afresh with the understanding developed through the **CONSTRCION** of something that works.
 - This is the good part of the Agile Philosophy
 - But instead, ensure that the way you go about developing systems
 - will always be first of all
 - intentionally
 - and intellectually
 - transparent
 - and manageable
 - Hence the insistence of excessive articulation and elaboration
 - write everything down that 1 can think of
 - preserve the context where ideas appear
 - do it in a way that they can then be refactored, reconsidered, elaborated
 - coherently, eventually complete and consistent
 - Very much in the spirit of the conversational REPL interaction
 - as you could o it in LISP in the old days
 - name your top level entry function, that name the subsystem
 - systematically described the intent in terms of sub functions and their combinations

- run the program on example input
 - and when it encounters a function not yet defined define it
- yes you will write a book first
 - for sure
 - c.f. <https://book.peergos.org/>
 - But do it always in some context where the ideas can be made to work
 - in however simple ways and parts
 - Design a system that supports this kind of working
- This will not be programming anymore
 - so not literate programming
 - but intentional affective formulations
- for this to work we need to create a system that has explicit self-organising, self-hyper-mapping capabilities
 - that is to say, it should be amenable to self-organization
 - and capable to bring to attention everything that is needed when you need it
- This is where the notation and the way of writing and storing and organizing searching etc need to be a coherent capable whole
 - instead of divide and conquer, capture the wholeness in a way that its implicate order will be emergent and refactorable as need arises
 - It is not Code that wins argument, but working systems!
 - *'Tain't what you do, but the way that u do it that's what's gets result*
 -
- /do: unity of information and action
 - name the capabilities you need and elaborate them in the complex of the info-communication-capability infrastructure they need to make them work as a whole
- /🌱
- /🌱
- /📄/index.html

[edit](#)

secret.share: [link](#) (requires password)

.purpose

- Create instance first exemplars of
 - dedicated personal spaces that include everything needed to make IT
 - minimal viable seed for the IndyWeb itself
 - work
 - local-first, personal-first, private, autonomous, lively, permanent and evergreen
 - that is to say once I had accessed any information using capabilities
 - that were shared to go with them,
 - I can then use to create/curate/share content that I engages with
 - respond, annotate, in=notate, converse a
 - and anything I engages with
 - I would own and have full control over it.
 - Control all yOur information flows,
 - along with all the capabilities I's need to work with them,
 - control yOur destiny.
- Assume that for any individual there will be interlinked circles of named
 - people
 - pursuits,
 - info-communication-capability **complexes**
 - that would empower people
 - to weave their own trust networks
 - connecting people, lifelong mutual learnings, interests and pursuits
 - relying on commons based, peer produced capabilities and networks
 - of trust mutual arising info-co-municipal|labora-tional
 - indra.net work/co-laboratory spaces
- These form their own multiplicities,
 - could be anywhere, all at once, lifelong
 - and only the nameless individual owner/creator and trusted named co-laborators
 - who themselves would have their
 - owned, singular complete view/access to all these things
 - A truly fractal mycelial network of
 - autonomous,
 - mutual arising,
 - co-
 - liberative
 - laborations,
 - symmathetic conversations in the medium of
 - Virtual Autopoietic Hyper Document Spaces
- *.cascading.views/morphic transformations-affordances*
- documents that can be an/in-notated
 - responded to in
 - private,
 - autonomous,
 - autopoietic
 - threaded conversations
 - real time or asynchronous as needed
 - which themselves are subject to further
 - elaborations,
 - affective morphic formulations,
 - comments,
 - responses
 - debates
 - arguments
 - all designed to simultaneously scaling
 - reach
 - salience,
 - synthesis,
 - symmathesy
 - in a word consilience
 - replace dumb, and oppressive, hierarchical, power structures
 - necessitated in the absence of capabilities
 - to maintain and manage information flows in networks of trust
 - with full provenance and audit trails that are mutually verifiable from within
 - not from a central controlling position, which cannot even be created
 - in this constellation
 - with self-organization
- where each an-in-notation with their implicate secret/permissioned/open
- *trans-share.links* where all sharing is verifiable recorded by all the participants
 - and when I accesses a named resource that actually creates a p2p channel between the user and the provider of the resource
- will
 - act as portals to these associative MEMPIEXes
- integral
 - omni-optional
 - multi-variant

- intentionally transparent and affective,
- morphic
- transactional
- transformational
 - interplays and
 - symmathetic workflows
- .

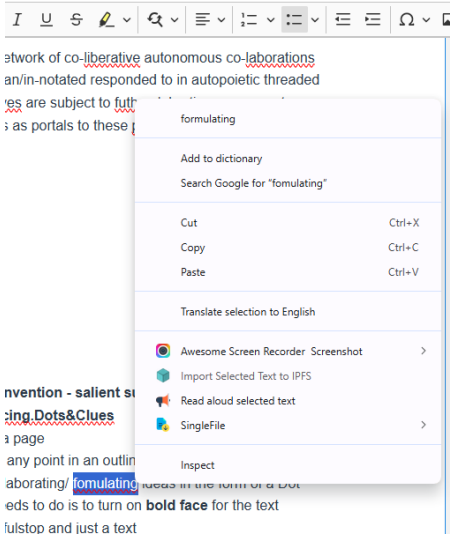
• .

🚩📖.update

- // new notational/do.how convention - salient summaries vs details
 - .in.line.notation - introducing Dots & Clues
 - In the top level of a page
 - as well as at any point in an outline
 - I can start elaborating/ formulating ideas as a Dot
 - .gloss=Dot
 - is the fundamental unit of structured writing
 - which comprising two major sturctural units
 - a title, name
 - and a nested list
 - such units are automatically assigned a local unique id
 - in an implicate MindGraph (or MindPlex)
 - wit the following properties
 - ☒.morphic.affordances-4-MindPlex.Dots
 - // use the blue check box emoji ☒
 - as an iconic qualifier, prefix for a clue
 - indicating that the item describes, con//tains a clue
 - to a to be developed feature or morphic affordanc
 - ^// an example for an in-conPlex in-line task description
 - so that I can describe in the conText a task
 - not in a separate task tracker/planner
 - from such indication task planning and tracking
 - functions can be derived and created
 - ^// an example for the dictum
 - anything that comes to mind
 - as I writes about anything
 - has to be recorded, named, qualified
 - in the context that they came to mind
 - The way the notation supports inline writing
 - creation and curation of Dots
 - ensures that these association are created explicitly
 - so that they can be acted upon later as required
 - they can appear at top level of a rich text page
 - or as a list item in an unordered list structure
 - at the point of saving a page
 - they are turned into Dot nodes in a mind graph
 - and the outline structure is replaced by a button to show/hide the outline on demand
 -
 - all I needs to do is to
 - turn on **bold face** input on in the editor for the text
 - type a fulstop and just a text
 - one that later on reflection can be reformulated into a
 - *ground,atomic.focmula* that can become the name of a Clue
 - following that stat an unordered list/outline structure
 - started composing salient summaries once I've doe a first draft of a mind dump
 - as I write-to-think
- .

.dislexia.strikes ~ spelling

- I find it difficult to see what is it that I miss spelled
 - find it hard to see that I did not type the r in formulating



// scatter dots on a page

- and turn them into trans-portals, high-resolution implicate intertwingularities

// map is the territory

- 🚩📖

// split named sections into two parts

- One for the territory the actual elaboration formulation in detail
- use the icon 🌿 **National park Emoji**
 - <https://aieemojifinder.com/?query=territory>
 - 🌿 🏠 🌿 🏠 🌿 🏠 🌿 🏠 🌿 🏠 🌿 🏠
- and for the map use this 📄
- or combinations of 🌿 📄
- etc

📖 .how.did.I.get.here

Having another crack at going all in with Peergos

main reasons are:

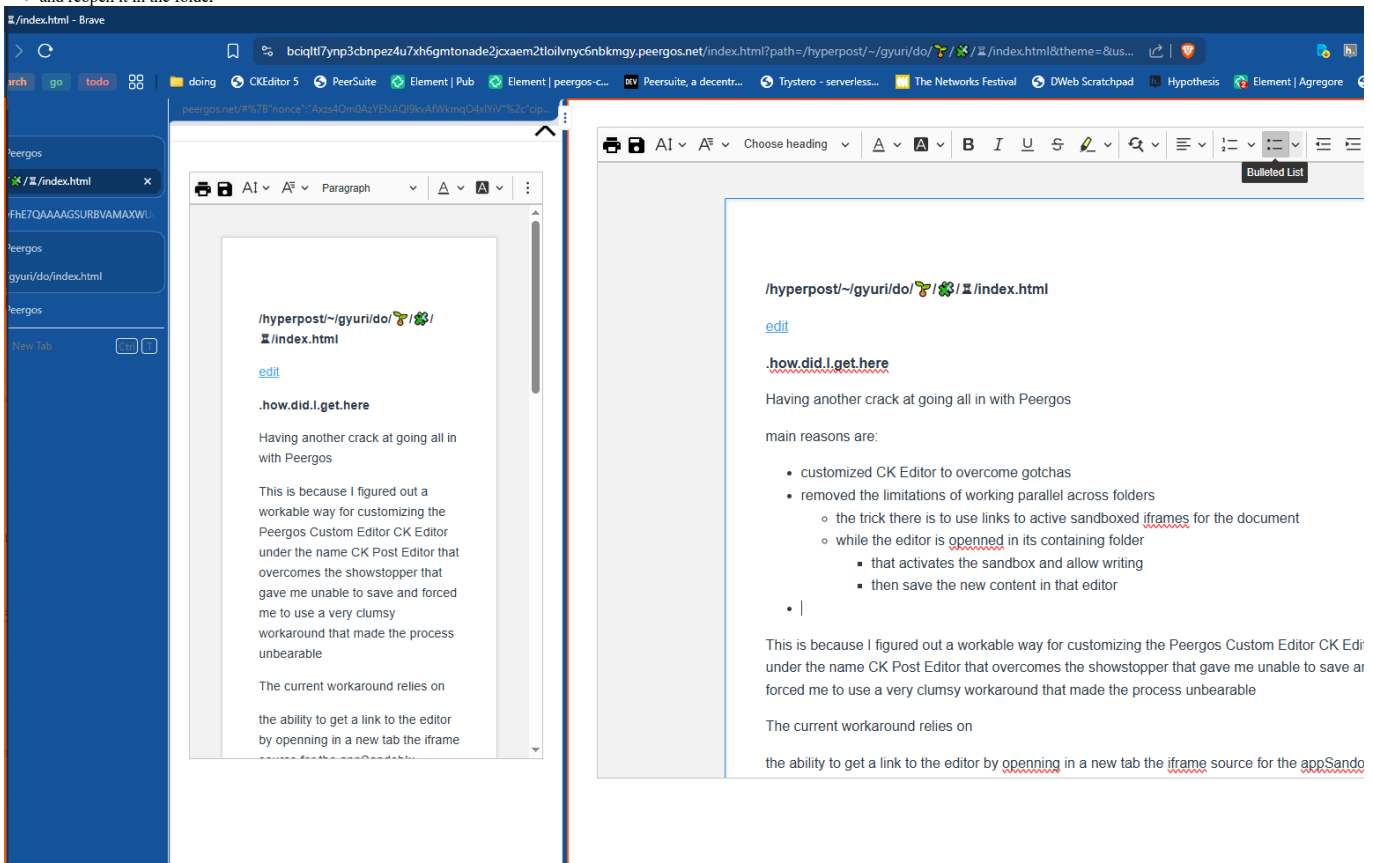
- customized CK Editor to overcome gotchas
 - removed the limitations of working parallel across folders
 - the trick there is to use links to active sandboxed iframes for the document
 - while the editor is opened in its containing folder
 - that activates the sandbox and allow writing
 - then save the new content in that editor
 - opportunity to inter(p)lay with MindGraph/Plex
 - as a personal/private, local-first, off-line first unitary indra-net-work-space
 - transfer oall the learnings and affordances that I explored with my own Rich Text Html Editor, use that as an alternative
 - Better still use it to annotate/innotate telegram like
 - two way linked Reply mechanism
 - omni-optionally use either editor
- Started experiment with secret folders for collaborating with people
 - who do not have a Peergos account just got a secret link to support
 - async information exchange
 - and permanent storage while all data that I engaged with
 - is available local first
 - offline first.
 - .
 - the hope is that if the shared folder also contains the apps that was used to create it
 - even people with no Peergos account will be able to save information in those folders that were shared with them

./ start of scaling synthesis

- Use utf8icons as qualiers for eleboration distinguish between detail work and salient synthesis and summaries for the same DotPlex

🌿.how.did.I.get.here

- Bellow I can see that the sandboxed frame is shown on the riht side of the split view
 - while the left side of the split view shows the editor tht activated that isandboxed iframe
 - is open
 - Even if we get an unable to save
 - we can close the editor opened in the enclosing folder
 - and reopen it in the folder

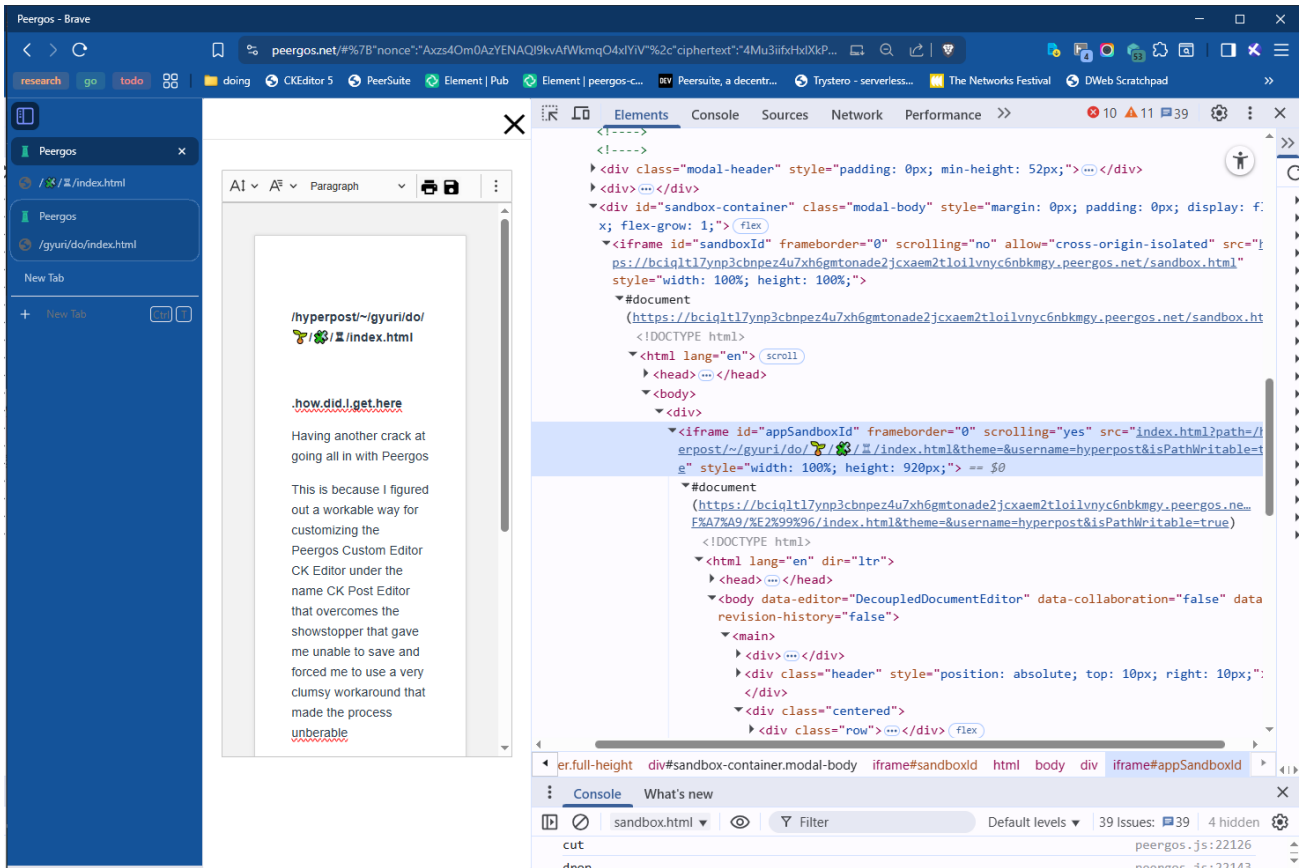


- Then saving succeeds
 - Still not as good as it could be
 - Next step is to save the page in MindGraph etc
 -

This is because I figured out a workable way for customizing the Peergos Custom Editor CK Editor under the name CK Post Editor that overcomes the showstopper that gave me unable to save and forced me to use a very clumsy workaround that made the process unbearable

The current workaround relies on

the ability to get a link to the editor by opening in a new tab the iframe source for the appSandbox



Brave Vertical Tabs in combination with the affordance opening link in a page in split view allow 1 to have access to the Peergos folder where the document is stored and see the document in the editor at the same time

Unfortunately the Title of a Peergos page is always just Peergos

Modified the code so that the title of the editor shows the last 3 elements of the document path

and this corms a block together with the Peergos folder shown on the left side of the split view

This gives a workeable setup. compatibility with the original CK Editor is maintained

which is good because I just had to fix a bug because the document sized exceeded the local storage size limit

as it tried to save it there

No problem in fact we need to use localforage perhaps or beter still

save automatically images into subfolders or IPFS or in Peergos

