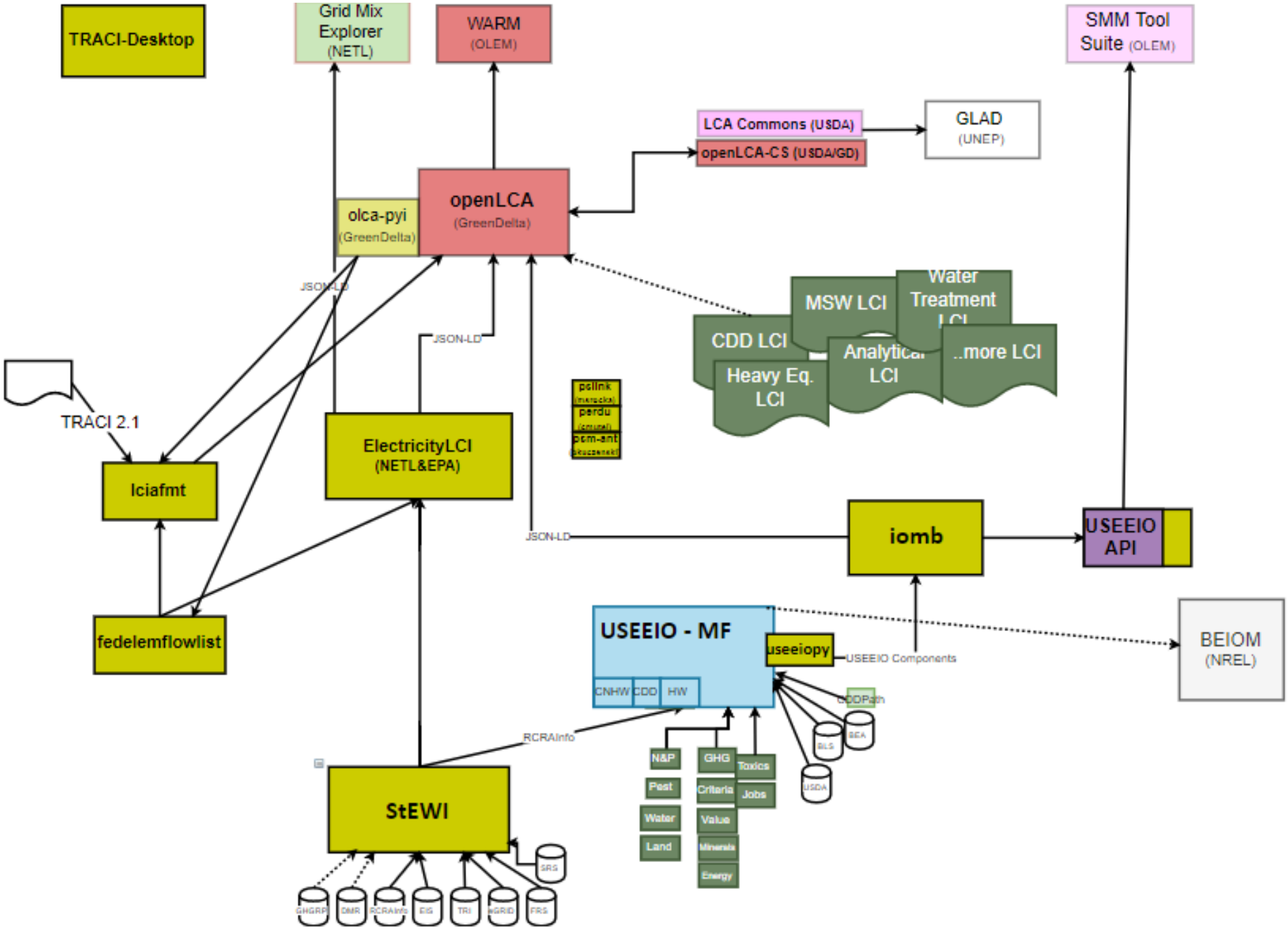
Tool Ecosystem

November 5, 2019

for EDAB

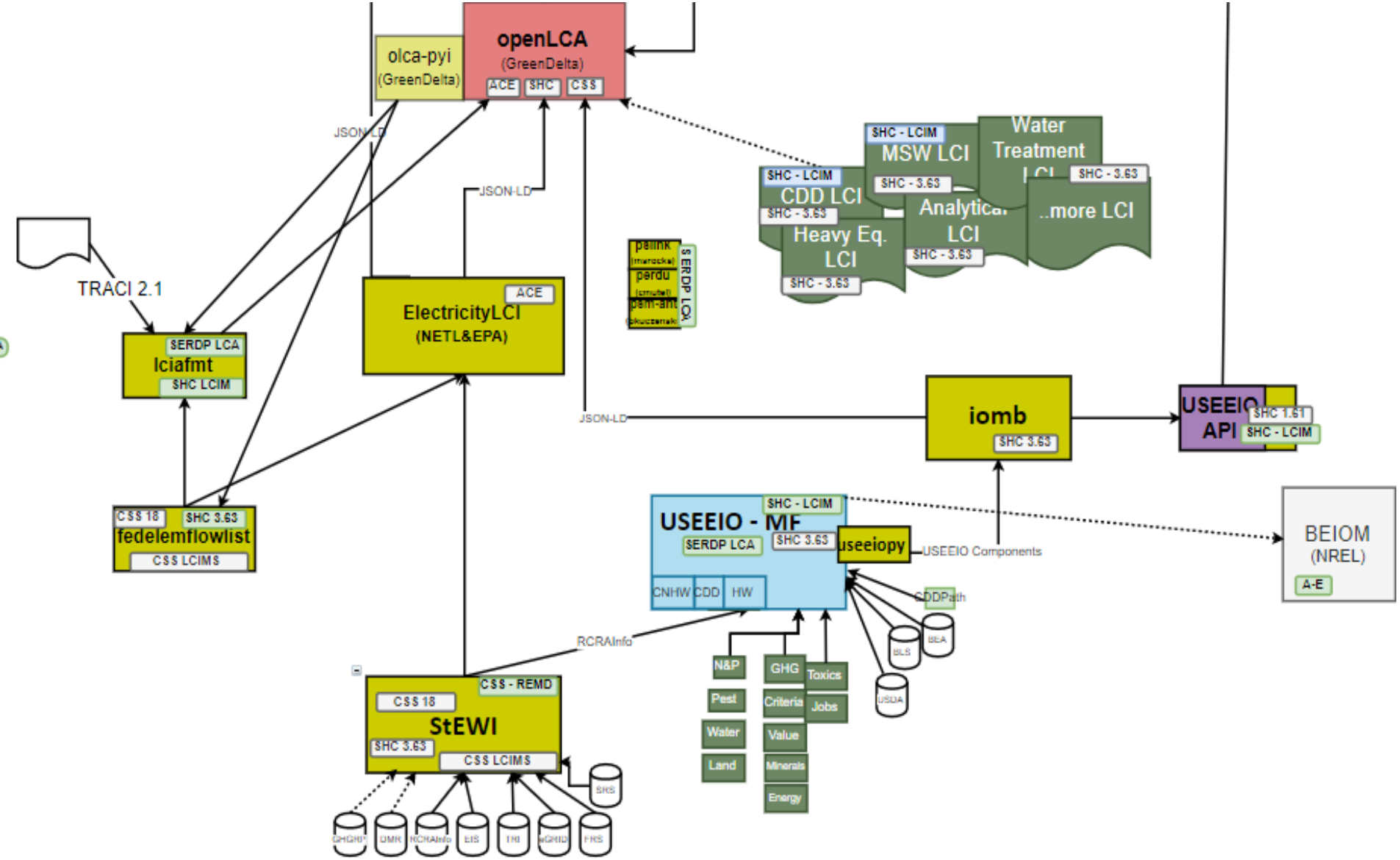
Wesley Ingwersen


TRACI-Desktop
Grid Mix Explorer (NETL)
WARM (OLEM)
SMM Tool Suite (OLEM)
LCA Commons (USDA)
openLCA-CS (USDA/GD)
GLAD (UNEP)
olca-pyi (GreenDelta)
openLCA (GreenDelta)
JSON-LD
JSON-LD
MSW LCI
Water Treatment LCI
CDD LCI
Heavy Eq. LCI
Analytical LCI
..more LCI
pollink
perdu
pom-ant
TRACI 2.1
lciafmt
ElectricityLCI (NETL&EPA)
JSON-LD
iomb
USEEIO API
fedelemflowlist
USEEIO - MF
CNHW
CDD
HW
useeiopy
USEEIO Components
BEIOM (NREL)
CDDPath
BLS
BEA
USDA
RCRAInfo
N&P
Pest
Water
Land
GHG
Criteria
Value
Minerals
Energy
Toxics
Jobs
StEWI
SRS
GHGRP
DMR
RCRAInfo
EIS
TRI
eGRID
FRS

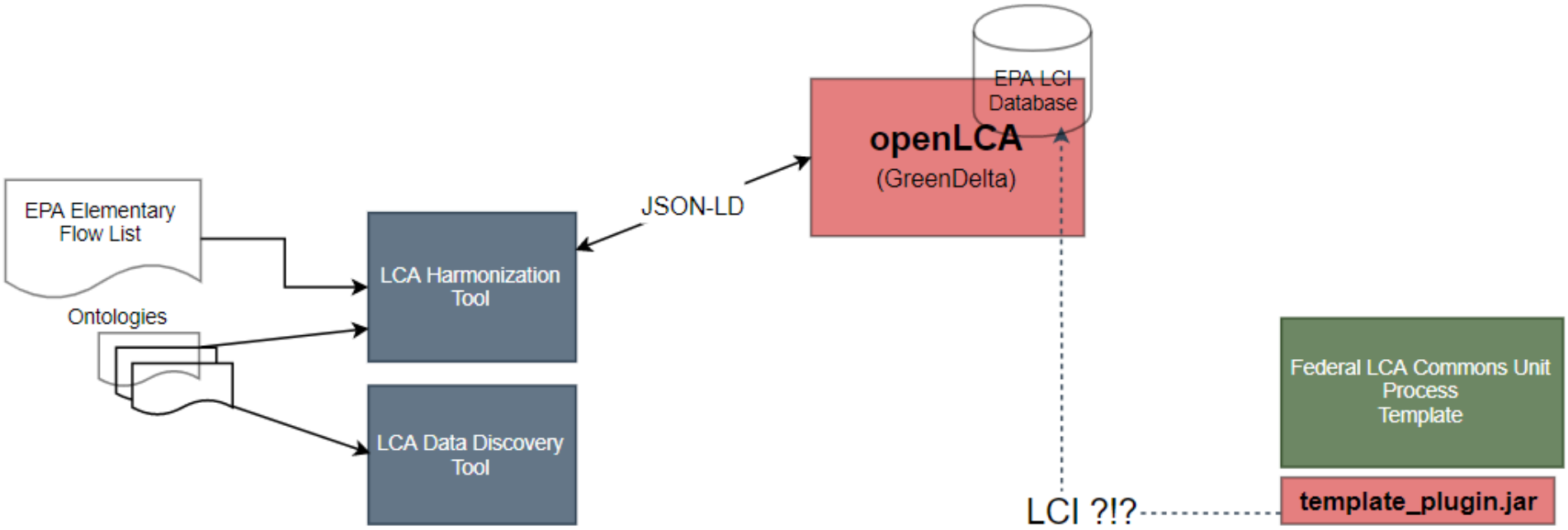
Our tools on github

Links for tools that we maintain

- Fedelemflowlist: https://github.com/usepa/Federal-LCA-Commons-Elementary-Flow-List/
- lciafmt: https://github.com/USEPA/LCIAformatter/
- USEEIO-MF: https://github.com/usepa/USEEIO
- iomb: https://github.com/USEPA/IO-Model-Builder
- USEEIO-API: https://github.com/USEPA/useeio_api/
- StEWI: https://github.com/USEPA/standardizedinventories
- ElectricityLCI: https://github.com/USEPA/electricityLCI/


olca-pyi
(GreenDelta)
openLCA
(GreenDelta)
ACE
$HC
CSS
JSON-LD
JSON-LD
$HC - LCIM
MSW LCI
$HC - 3.63
Water
Treatment
$HC - 3.63
$HC - LCIM
CDD LCI
$HC - 3.63
Analytical
LCI
$HC - 3.63
..more LCI
Heavy Eq.
LCI
$HC - 3.63
pelink
perdu
psm-ant
SERDP LCA
TRACI 2.1
ACE
ElectricityLCI
(NETL&EPA)
$ERDP LCA
lciafmt
$HC LCIM
JSON-LD
iomb
$HC 3.63
USEEIO
API
$HC 1.61
$HC - LCIM
CSS 18
$HC 3.63
fedelemflowlist
CSS LCIMS
$HC - LCIM
USEEIO - MF
$ERDP LCA
$HC 3.63
useeiopy
USEEIO Components
BEIOM
(NREL)
A-E
CNHW
CDD
HW
CDDPath
RCRAInfo
N&P
GHG
Toxics
Pest
Criteria
Jobs
Water
Value
Land
Minerals
Energy
BLS
BEA
USDA
CSS - REMD
CSS 18
StEWI
$HC 3.63
CSS LCIMS
SNS
GHGRP
DMR
RCRAInfo
EIS
TRI
eGRID
FRS

Extinction: What?



Extinction – Why?

- Architecture was external
- Architecture was proprietary
- Development skill was ONLY external
- No connections to other tools
- No more direct funding
- Story: LCA-HT

Recent History of LCA(?) Research

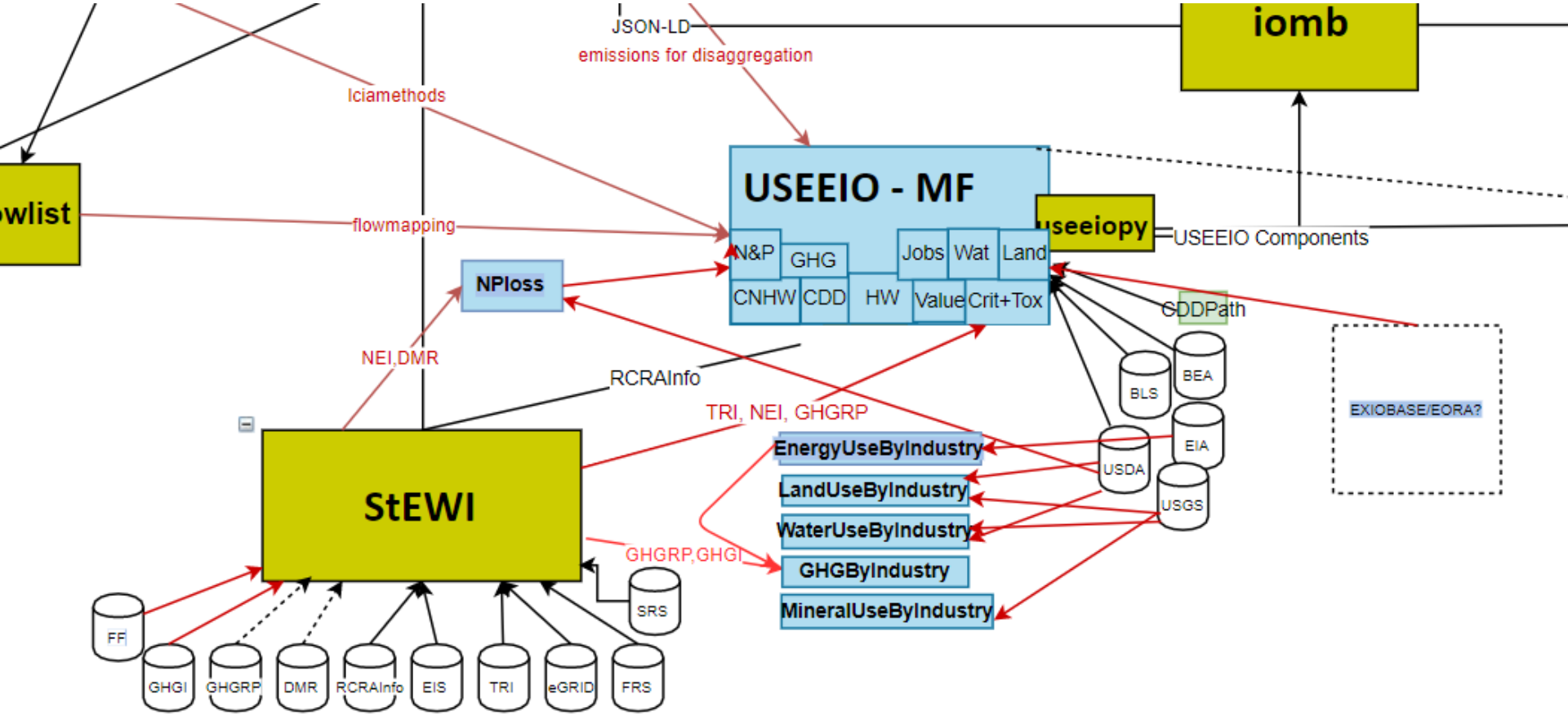
A while back…

- LCA Guidance
- LCA Studies
- Internal LCI database and SOP
- Produce process-based, gate-to-gate LCI
- LCIA method (TRACI)
- LCI method development

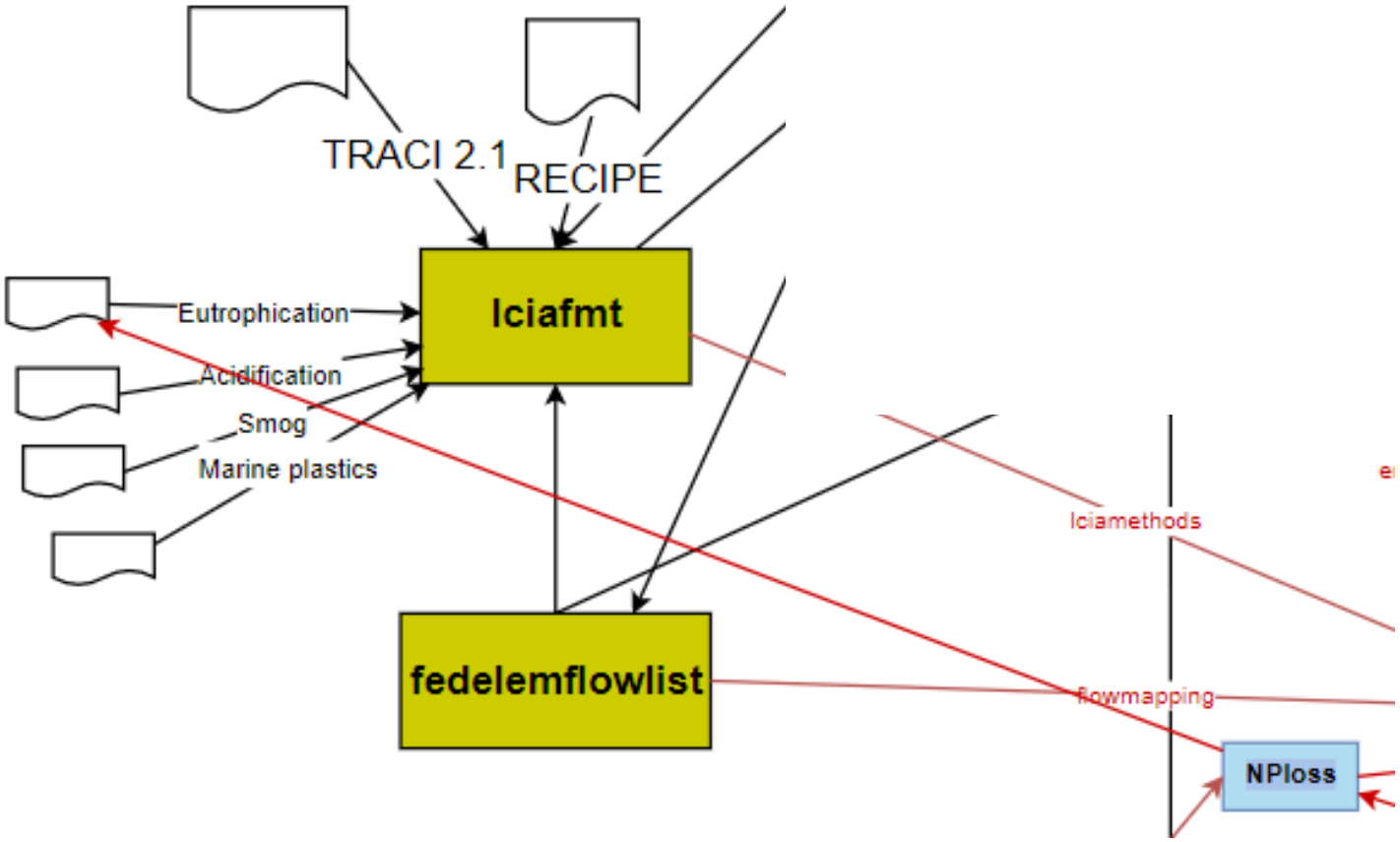
More recently …

- Produce basic LCA components (DQ, Flow list)
- Maintain 1 full LCA model (USEEIO), LCIA method (TRACI)
- Release modeling (chemical production/waste management)
- Material/waste use, generation and disposition

USEEIO Near-Term Plans



LCIA – Near term



Why Tools?

- Preservation of knowledge
- Availability of methods
- Reproducible
- Good science/engineering practice

Tools - Design

- Modularity and connectivity
- Design for internal and skilled external users
- Use platform independent, open source toolkit
- Create a transparent, reproducible data flow
- Produce standard formats
- Make it flexible
- Use standard operating procedures (SOPs)
- Write and use tests
- Package It

Tools - Management

- git(hub) management and collaboration –Other collaboration tools – JIRA
- A team/organizational job
- Build partnerships
- Story: ElectricityLCI

Tools - Getting Started

(assuming there is a RAP or other justification)

- When should it become a tool?

  Story: Federal LCA Commons Elementary Flow List

- Search around for existing code projects – github, internal (RCS), CRAN/PyPI

- Check with colleagues on possible related work

- Bring partners on board

- Write functional and technical requirements

Modularity

- Large scope tools are harder to manage
- Rich resources get hidden

Using git hosts - github/bitbucket

- Highly effective collaborative workflows
- If tools go dormant, at least they persist on the repo

Tips

- Add partners to repos
- Let external contributors fork and pull
- Use branching for all new functionality
- Make ‘Releases’ for stable versions for use/testing

Design for internal and skilled external users

- By us, largely for us
- No GUIs
- Avoid web hosting
- No lay user interaction for development
- Simplified documentation

Transparent, reproducible data flow and standard formats

- Show URL/location/methods for obtaining original resource

- Capture metadata in standard format

-- Use JSON or YAML

- Story: StEWI

Flexible tools

- Allow for multiple data sources/methods and create multiple models
- Allow for multiple outputs
- Story: USEEIO