Arinas Platform

A visible architecture made easy with

DConf ‘25
London
August 21, 2025

Ľudovít Lučenič [Lyoodoveet Loochenyich]
ludovit.lucenic@digital-orchestra.sk

Digital Orchestra, s.r.o.
Simple things fast. Complex things simple.

Main message

- **The D language** is **efficient** from a **computational** and **systems** perspective
- **Arinas Platform** builds on this foundation
  to deliver **efficiency** in the software development **process**
- Moreover, if you choose to compile its
  LeS language to **native D code**, it can yield
  a theoretical **synergy** of both strengths in the end

Our domain

uncertainty
costs
Arinas
information
interpreter
efficiency
tools
LeS
LeS
complexity
visibility
platform
D
model
language
chaos
software
entropy
uncertainty
systems
tree
development
efficiency
compilation
order
structure
requirements
chaos
model
D
tools
platform
order
tree
systems
Dengineering
architecture
architecture
platform
engineering
information
complexity
development
costs
compilation

Process efficiency

- How do we measure **efficiency**? By costs
  - time, resources, money, opportunities, ...
- What **costs** the most? Uncertainty
- What is the **cause** of uncertainty? **Entropy**
- How does entropy show up? As **complexity**

Process efficiency – cont’d

- How do we **approach** software complexity?
  - With models, decomposition & **architecture**
- What is the process efficiency aspect of software architecture?
  - The ability to **answer** architectural **questions** quickly and consistently
- What enables that?

Visible software architecture

Software architecture

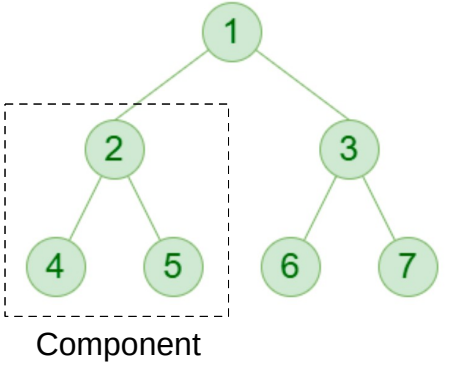
- **concepts** and features
- components, relationships
- design and evolution **principles**
- **decisions** that shape system structure and set constraints or guide its all other designs

Software architecture – a city map

- Imagine a system as a city:
  - roads = data flows
  - districts = domains
  - buildings = components
- Seeing its architecture is like having its city map
- If the map is clear, decisions are faster
- The Arinas Platform makes that map clear

Arinas Platform

- A development and runtime environment based on tree structures
- It represents and interprets the program as a tree – along with its inputs, outputs, data, code, and architecture
- Control flow via tree traversal



Arinas Platform

- **LeS language** – a programming language designed for tree manipulation
- Main use cases: web and portal applications
- Virtually any event-driven software can be implemented using this concept

A tree ?

Structure
System
Information

Arinas – Concepts and Paradigm

- **Pattern** (a module instance)
  - a tree with a module **type**, **attributes** (name, description, …), and **subordinate** patterns governed by the module's specification
- **Package** (a module set)
  - defined by struct type (package), location, name

Arinas – Concepts and Paradigm

- **Pattern example**

```
ret{
  /__type = "content/portal/default";
  /name = "Application portal";
  /login = "free";
  /template = "↔";
  /__sub/main{
    /__type = "content/main";
    /name = "Main page";
    /template = "↔";
    /__sub{
      /cmdEditor{
        /__type = "content/editor";
        /name = "Input tree editor";
        /label = "INPUT TREE";
        /defaultCode = "↔";
        /__sub = empty;
      }
      /codeEditor{
        /__type = "content/editor";
        /name = "Code editor";
        /label = "<b>ONLINE EDITOR</b>";
        /defaultCode = "↔";
        /__sub = empty;
      }
      /retEditor{
        /__type = "content/editor";
        /name = "Output tree editor";
        /label = "OUTPUT TREE";
        /defaultCode = "ret = empty;";
        /__sub = empty;
      }
    }
  }
}
```

Arinas – Concepts and Paradigm

- **Dimensions**
  - spatial, object hierarchy, pattern subordination
- **Module** (a component)
  - struct type (module), **location** in the module set (i.e. module type), name, **parent** module type, private data and pattern definitions, **operations**, **attributes** and the **subconfiguration** definition, versioning and upgrade information, submodule definitions

Arinas – Concepts and Paradigm

- **Module example**

```
ret{
  /__struct = "module";
  /parent = "/base";
  /name = "CMS root module";
  /description = "";
  /patterns = empty;
  /data = empty;
  /operations/default_{
    /type/les = 1.0;
    /code = "↔";
    /modified = #2025-02-19T15:12:19.8484243#;
  }
  /config{
    /attr/default/name = "Default content portal";
    /allowed = empty;
    /default = empty;
  }
  /version = empty;
  /__sub = empty;
}
```

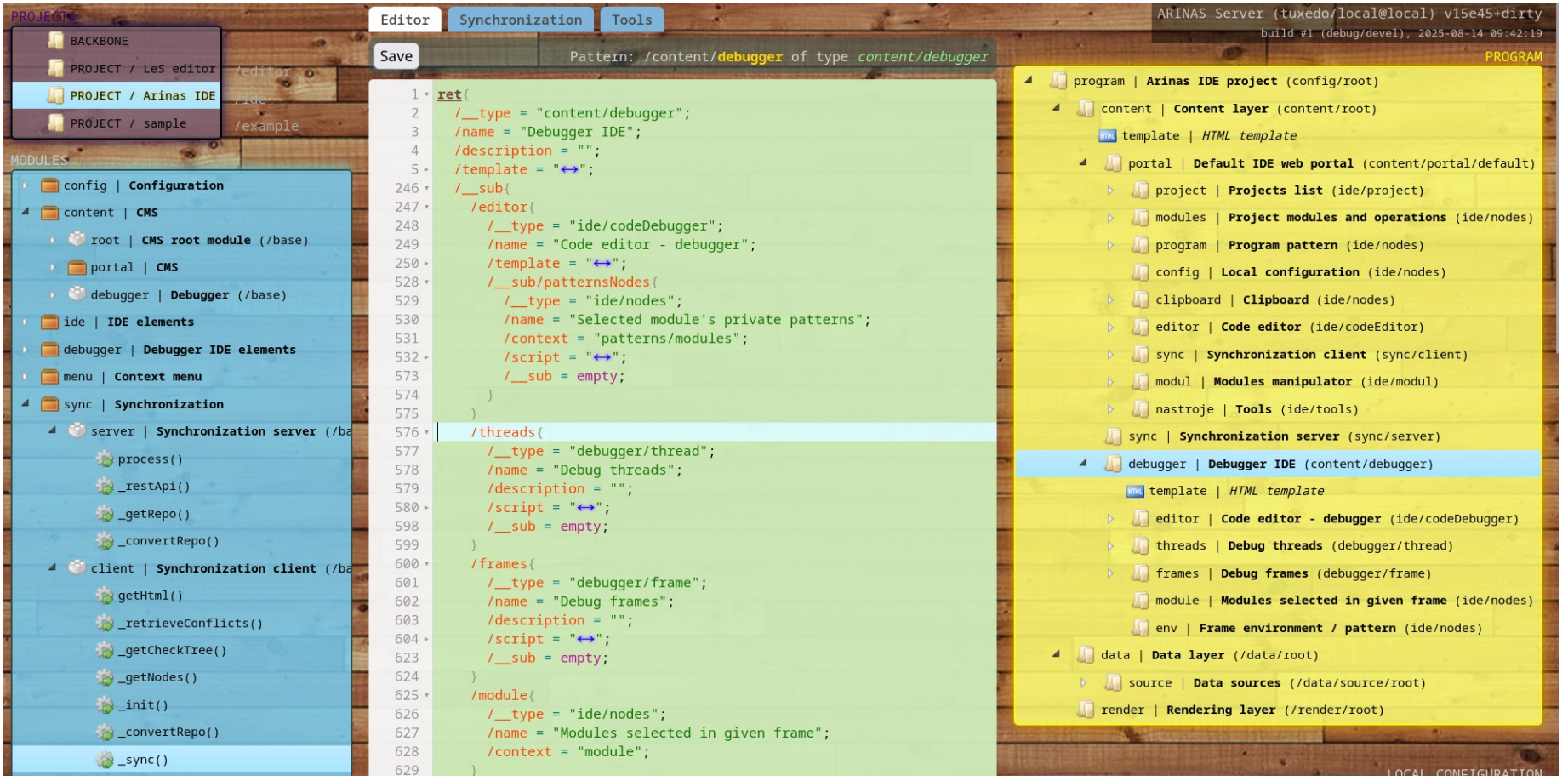
Arinas – Concepts and Paradigm

- **Module operation**

Module definitions and **patterns** are interpreted as trees. **Operations** are executed. Everything is written and represented **uniformly** in LeS code.

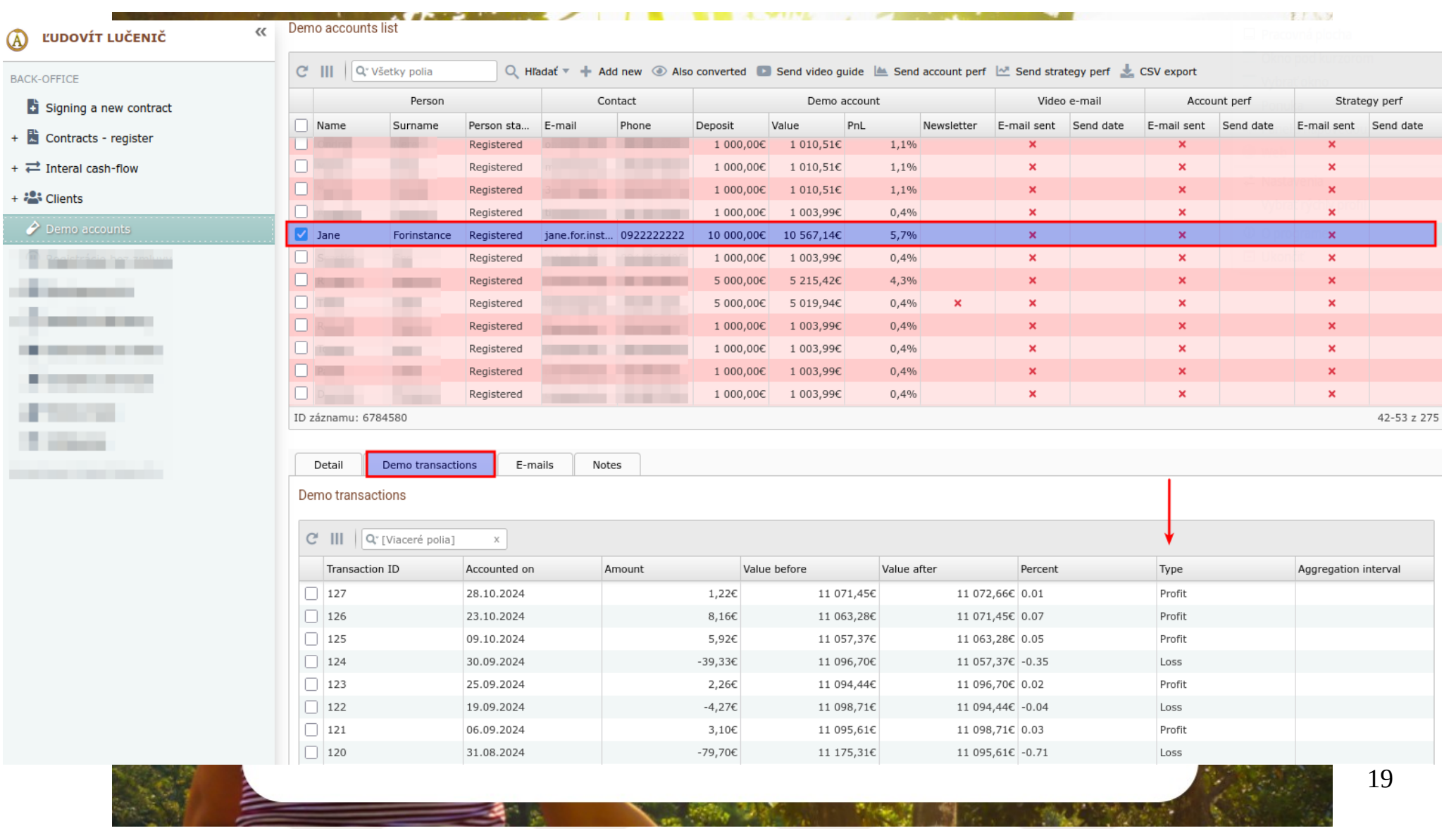
```
// synchronize current project with repository
var/branch = is.stringval(cmd/input/branch) ? cmd/input/branch : "MAIN";
var/branch = var/branch.split("#");

var/request{
  /command/sync{
    /context = ses/currentProject;
    /branch = var/branch[0];
    /revision = var/branch[1];
    /data = cmd/project;
  }
} = cmd/request;

var/response = net.http.send{
  /method = "POST";
  /url = cmd/host ~ "/sync";
  /content = node.toBase64(var/request);
};
var/response/response = node.fromBase64(var/response/response);
// extract conflicting patches (different patches at the same context)
// and filter them out from /client and /server trees into /conflicts tree
var/filterConflictsIns{ /showSrc = cmd/showSrc; } = cur;
ses/sync/data = @var/filterConflictsIns._retrieveConflicts(var/response/response/data);

// we log here the received patchsets until we have usable IDE in this regard
for (var/response/response/data) {
  // we transform the structure to the bare minimum for diagnostic purposes
  var/datalog[key]{
    [src/type][src/context] = null;
    /patch = src/patch;
    if ((?cmd/showSrc) && (src/type/upd || src/type/ord)) {
      /src = src/src;
      /dst = src/dst;
    }
  } in val;
}
info(var/datalog);

ret = var/response;
```

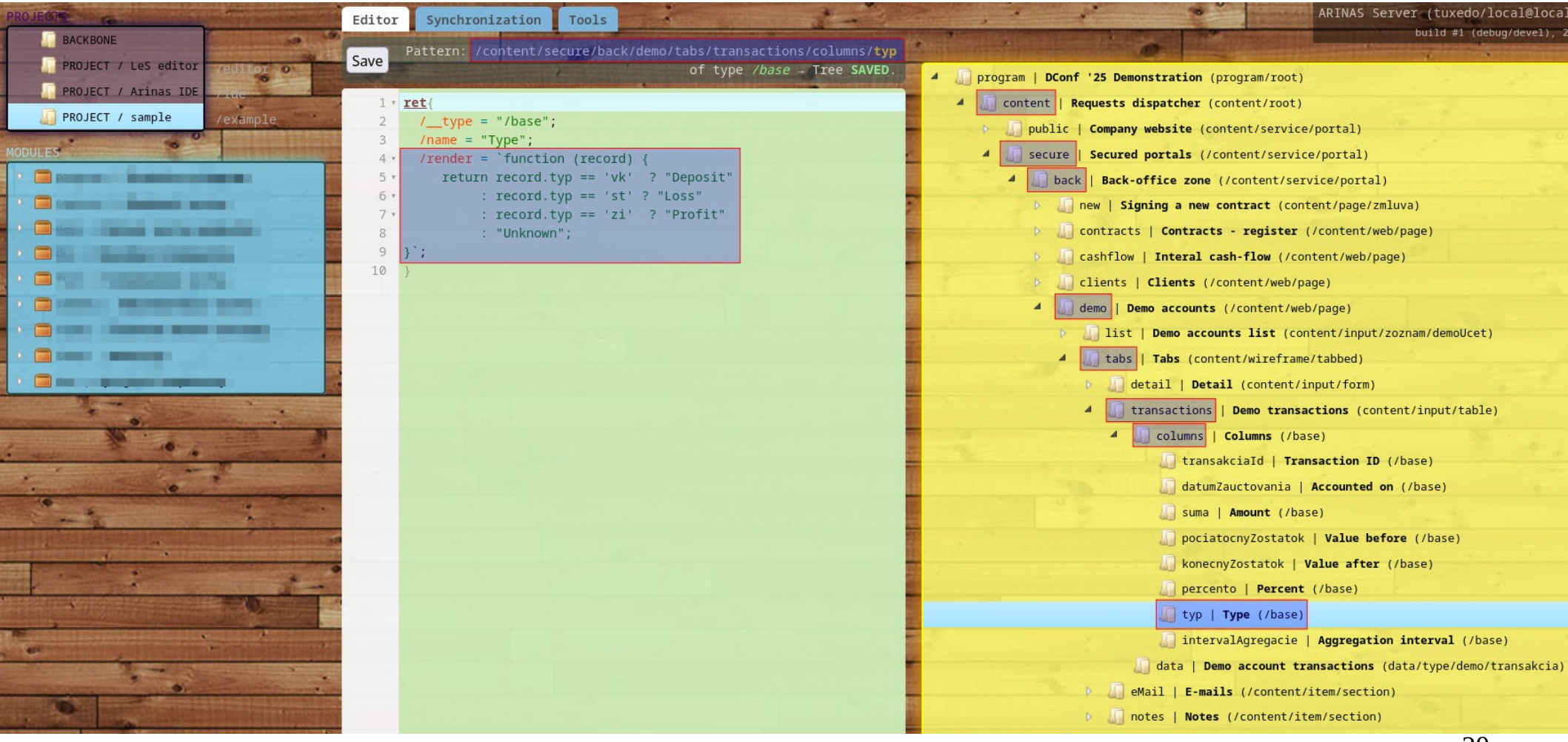
Arinas Platform - IDE

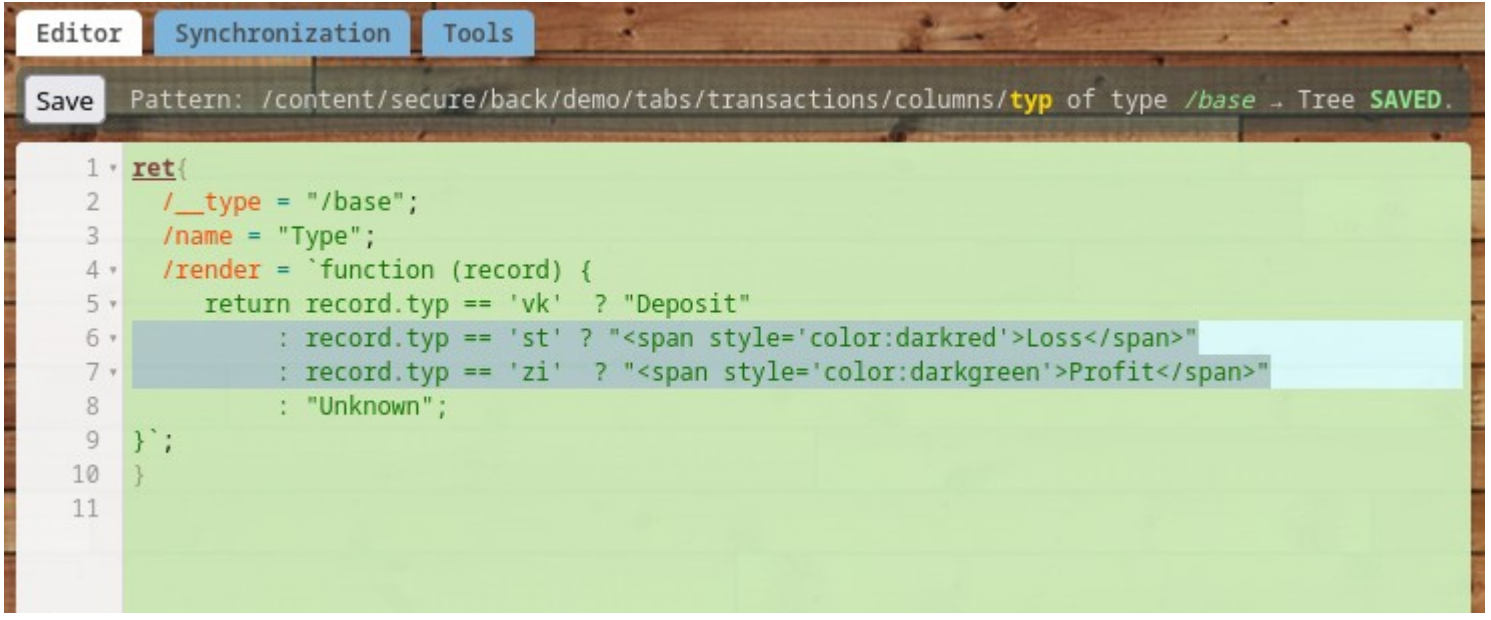


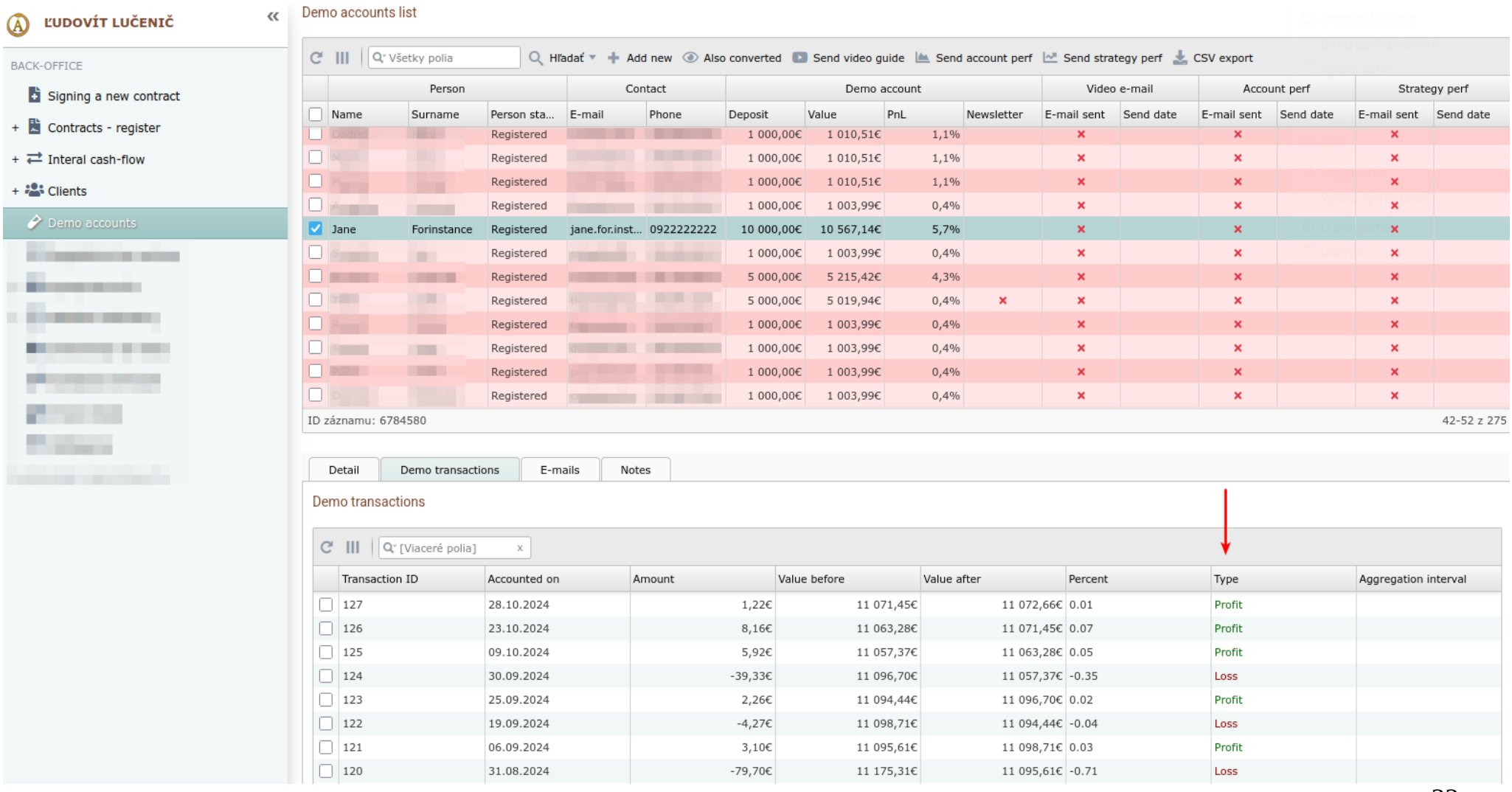
Arinas Platform – Live Demo

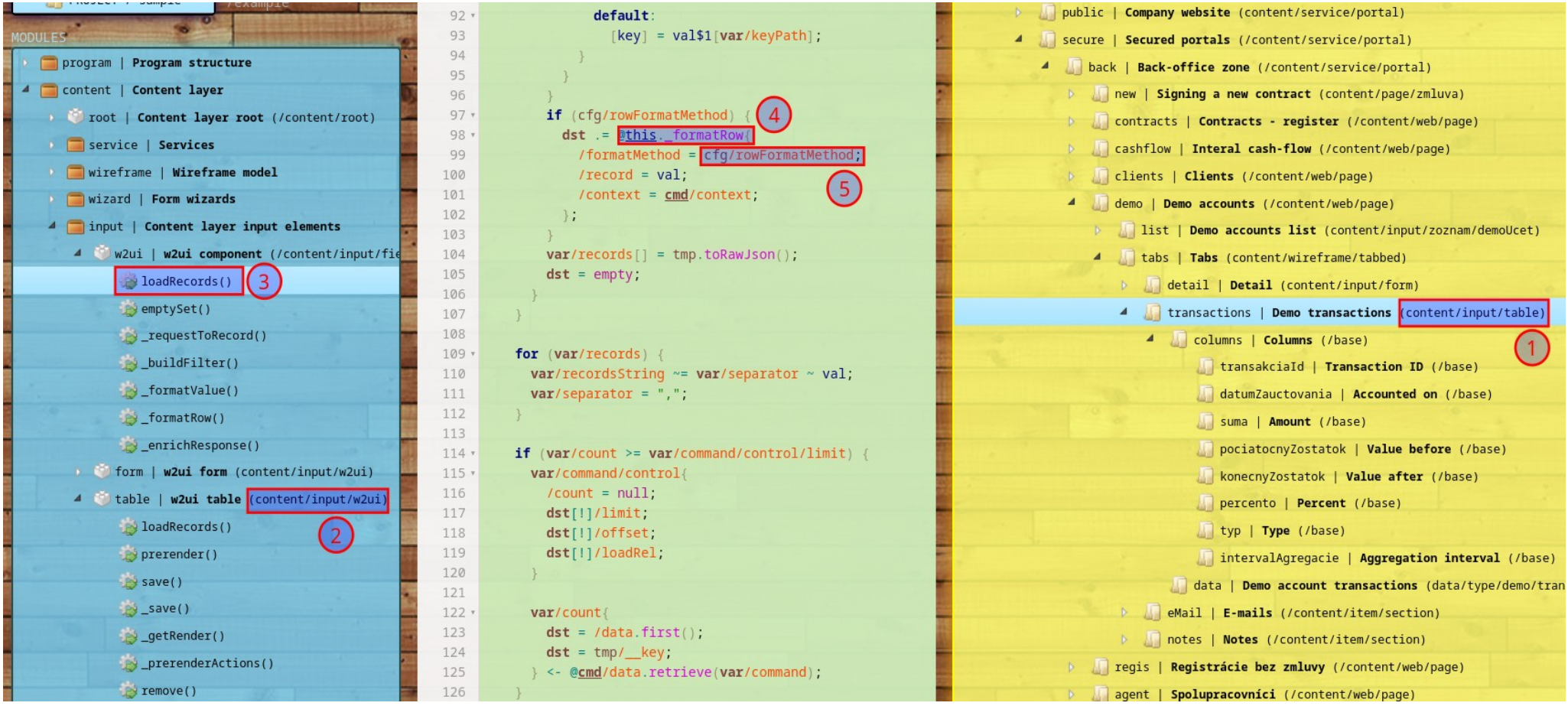
- Live examples
  - locating a component
  - implementing requirements at various levels
  - instantiation and use in different contexts
- Architectural levels
  - top level
  - layer
  - component
  - element / function
  - configuration

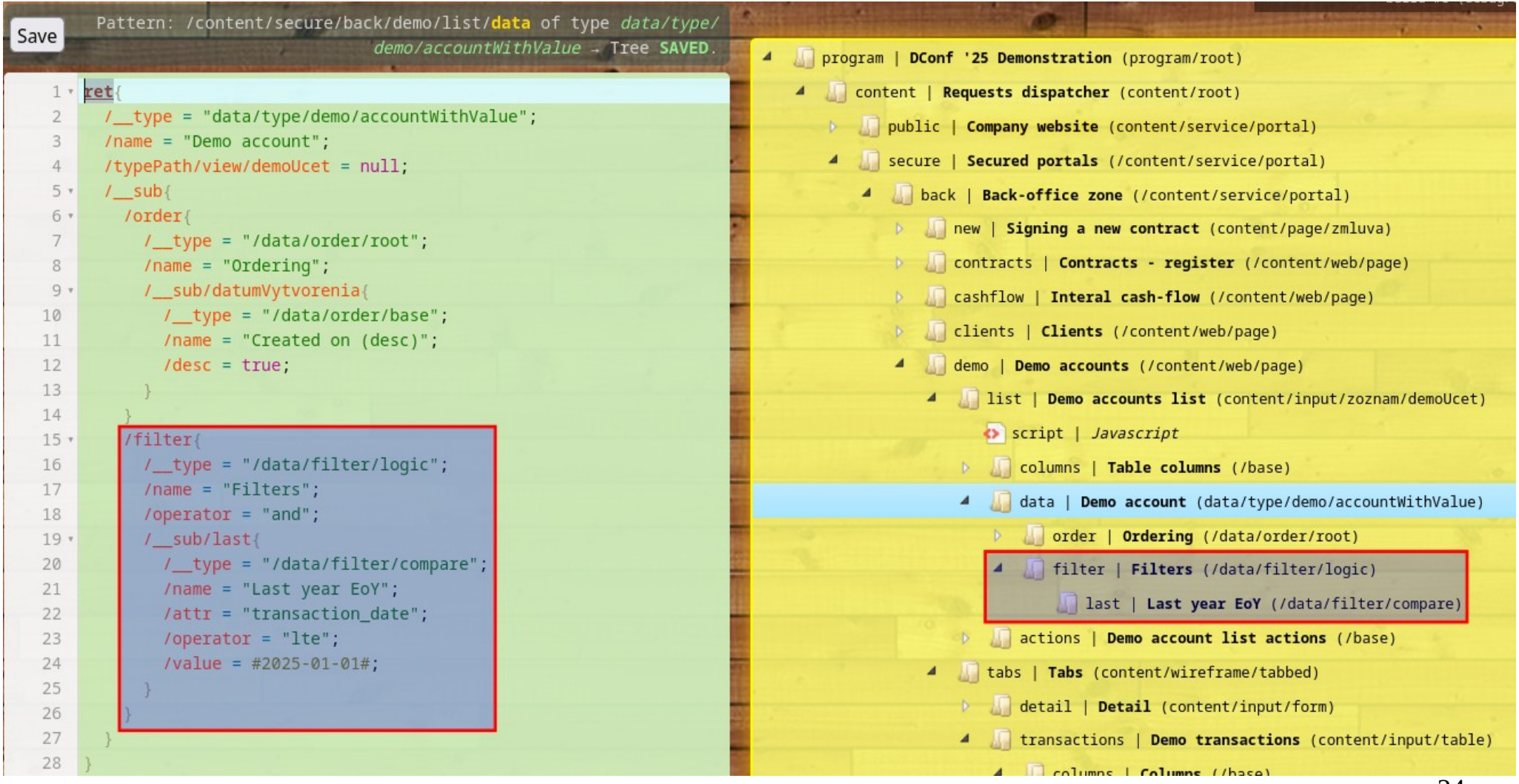

ĽUDOVÍT LUČENIČ
BACK-OFFICE
Signing a new contract
Contracts - register
Interal cash-flow
Clients
Demo accounts
Demo accounts list
Všetky polia
Hľadať
Add new
Also converted
Send video guide
Send account perf
Send strategy perf
CSV export
Person
Contact
Demo account
Video e-mail
Account perf
Strategy perf
Name
Surname
Person sta...
E-mail
Phone
Deposit
Value
PnL
Newsletter
E-mail sent
Send date
Registered
1 000,00€
1 010,51€
1,1%
1 003,99€
0,4%
Jane
Forinstance
jane.for.inst...
0922222222
10 000,00€
10 567,14€
5,7%
5 000,00€
5 215,42€
4,3%
5 019,94€
ID záznamu: 6784580
42-53 z 275
Detail
Demo transactions
E-mails
Notes
Demo transactions
[Viaceré polia]
Transaction ID
Accounted on
Amount
Value before
Value after
Percent
Type
Aggregation interval
127
28.10.2024
1,22€
11 071,45€
11 072,66€
0.01
Profit
126
23.10.2024
8,16€
11 063,28€
11 071,45€
0.07
Profit
125
09.10.2024
5,92€
11 057,37€
11 063,28€
0.05
Profit
124
30.09.2024
-39,33€
11 096,70€
11 057,37€
-0.35
Loss
123
25.09.2024
2,26€
11 094,44€
11 096,70€
0.02
Profit
122
19.09.2024
-4,27€
11 098,71€
11 094,44€
-0.04
Loss
121
06.09.2024
3,10€
11 095,61€
11 098,71€
0.03
Profit
120
31.08.2024
-79,70€
11 175,31€
11 095,61€
-0.71
Loss


PROJECTS
BACKBONE
PROJECT / LeS editor
/editor
PROJECT / Arinas IDE
/ide
PROJECT / sample
/example
MODULES
Editor
Synchronization
Tools
ARINAS Server (tuxedo/local@local
build #1 (debug/devel), 2
Save
Pattern: /content/secure/back/demo/tabs/transactions/columns/typ
of type /base - Tree SAVED.
1 ret{
2 /__type = "/base";
3 /name = "Type";
4 /render = `function (record) {
5 return record.typ == 'vk' ? "Deposit"
6 : record.typ == 'st' ? "Loss"
7 : record.typ == 'zi' ? "Profit"
8 : "Unknown";
9 }`;
10 }
program | DConf '25 Demonstration (program/root)
content | Requests dispatcher (content/root)
public | Company website (content/service/portal)
secure | Secured portals (/content/service/portal)
back | Back-office zone (/content/service/portal)
new | Signing a new contract (content/page/zmluva)
contracts | Contracts - register (/content/web/page)
cashflow | Interal cash-flow (/content/web/page)
clients | Clients (/content/web/page)
demo | Demo accounts (/content/web/page)
list | Demo accounts list (content/input/zoznam/demoUcet)
tabs | Tabs (content/wireframe/tabbed)
detail | Detail (content/input/form)
transactions | Demo transactions (content/input/table)
columns | Columns (/base)
transakciaId | Transaction ID (/base)
datumZauctovania | Accounted on (/base)
suma | Amount (/base)
pociatocnyZostatok | Value before (/base)
konecnyZostatok | Value after (/base)
percento | Percent (/base)
typ | Type (/base)
intervalAgregacie | Aggregation interval (/base)
data | Demo account transactions (data/type/demo/transakcia)
eMail | E-mails (/content/item/section)
notes | Notes (/content/item/section)


Editor
Synchronization
Tools
Save
Pattern: /content/secure/back/demo/tabs/transactions/columns/typ of type /base → Tree SAVED.
ret{
/__type = "/base";
/name = "Type";
/render = `function (record) {
return record.typ == 'vk' ? "Deposit"
: record.typ == 'st' ? "Loss"
: record.typ == 'zi' ? "Profit"
: "Unknown";
}`;
}


ĽUDOVÍT LUČENIČ
BACK-OFFICE
Signing a new contract
Contracts - register
Interal cash-flow
Clients
Demo accounts
Demo accounts list
Všetky polia
Hľadať
Add new
Also converted
Send video guide
Send account perf
Send strategy perf
CSV export
Person
Contact
Demo account
Video e-mail
Account perf
Strategy perf
Name
Surname
Person sta...
E-mail
Phone
Deposit
Value
PnL
Newsletter
E-mail sent
Send date
Registered
1 000,00€
1 010,51€
1,1%
1 003,99€
0,4%
Jane
Forinstance
jane.for.inst...
0922222222
10 000,00€
10 567,14€
5,7%
5 000,00€
5 215,42€
4,3%
5 019,94€
ID záznamu: 6784580
42-52 z 275
Detail
Demo transactions
E-mails
Notes
Demo transactions
[Viaceré polia]
Transaction ID
Accounted on
Amount
Value before
Value after
Percent
Type
Aggregation interval
127
28.10.2024
1,22€
11 071,45€
11 072,66€
0.01
Profit
126
23.10.2024
8,16€
11 063,28€
11 071,45€
0.07
Profit
125
09.10.2024
5,92€
11 057,37€
11 063,28€
0.05
Profit
124
30.09.2024
-39,33€
11 096,70€
11 057,37€
-0.35
Loss
123
25.09.2024
2,26€
11 094,44€
11 096,70€
0.02
Profit
122
19.09.2024
-4,27€
11 098,71€
11 094,44€
-0.04
Loss
121
06.09.2024
3,10€
11 095,61€
11 098,71€
0.03
Profit
120
31.08.2024
-79,70€
11 175,31€
11 095,61€
-0.71
Loss


MODULES
program | Program structure
content | Content layer
root | Content layer root (/content/root)
service | Services
wireframe | Wireframe model
wizard | Form wizards
input | Content layer input elements
w2ui | w2ui component (/content/input/fie
loadRecords()
3
emptySet()
_requestToRecord()
_buildFilter()
_formatValue()
_formatRow()
_enrichResponse()
form | w2ui form (content/input/w2ui)
table | w2ui table (content/input/w2ui)
loadRecords()
2
prerender()
save()
_save()
_getRender()
_prerenderActions()
remove()
default:
[key] = val$1[var/keyPath];
if (cfg/rowFormatMethod) {
4
dst .= @this._formatRow(
/formatMethod = cfg/rowFormatMethod;
/record = val;
/context = cmd/context;
5
};
var/records[] = tmp.toRawJson();
dst = empty;
for (var/records) {
var/recordsString ~= var/separator ~ val;
var/separator = ",";
if (var/count >= var/command/control/limit) {
var/command/control{
/count = null;
dst[!]/limit;
dst[!]/offset;
dst[!]/loadRel;
var/count{
dst = /data.first();
dst = tmp/__key;
} <- @cmd/data.retrieve(var/command);
public | Company website (content/service/portal)
secure | Secured portals (/content/service/portal)
back | Back-office zone (/content/service/portal)
new | Signing a new contract (content/page/zmluva)
contracts | Contracts - register (/content/web/page)
cashflow | Interal cash-flow (/content/web/page)
clients | Clients (/content/web/page)
demo | Demo accounts (/content/web/page)
list | Demo accounts list (content/input/zoznam/demoUcet)
tabs | Tabs (content/wireframe/tabbed)
detail | Detail (content/input/form)
transactions | Demo transactions (content/input/table)
columns | Columns (/base)
1
transakciaId | Transaction ID (/base)
datumZauctovania | Accounted on (/base)
suma | Amount (/base)
pociatocnyZostatok | Value before (/base)
konecnyZostatok | Value after (/base)
percento | Percent (/base)
typ | Type (/base)
intervalAgregacie | Aggregation interval (/base)
data | Demo account transactions (data/type/demo/tran
eMail | E-mails (/content/item/section)
notes | Notes (/content/item/section)
regis | Registrácie bez zmluvy (/content/web/page)
agent | Spolupracovníci (/content/web/page)


Save
Pattern: /content/secure/back/demo/list/data of type data/type/demo/accountWithValue → Tree SAVED.
ret{
/__type = "data/type/demo/accountWithValue";
/name = "Demo account";
/typePath/view/demoUcet = null;
/__sub{
/order{
/__type = "/data/order/root";
/name = "Ordering";
/__sub/datumVytvorenia{
/__type = "/data/order/base";
/name = "Created on (desc)";
/desc = true;
}
}
/filter{
/__type = "/data/filter/logic";
/name = "Filters";
/operator = "and";
/__sub/last{
/__type = "/data/filter/compare";
/name = "Last year EoY";
/attr = "transaction_date";
/operator = "lte";
/value = #2025-01-01#;
}
}
}
}
program | DConf '25 Demonstration (program/root)
content | Requests dispatcher (content/root)
public | Company website (content/service/portal)
secure | Secured portals (/content/service/portal)
back | Back-office zone (/content/service/portal)
new | Signing a new contract (content/page/zmluva)
contracts | Contracts - register (/content/web/page)
cashflow | Interal cash-flow (/content/web/page)
clients | Clients (/content/web/page)
demo | Demo accounts (/content/web/page)
list | Demo accounts list (content/input/zoznam/demoUcet)
script | Javascript
columns | Table columns (/base)
data | Demo account (data/type/demo/accountWithValue)
order | Ordering (/data/order/root)
filter | Filters (/data/filter/logic)
last | Last year EoY (/data/filter/compare)
actions | Demo account list actions (/base)
tabs | Tabs (content/wireframe/tabbed)
detail | Detail (content/input/form)
transactions | Demo transactions (content/input/table)

Interpreter | Code interpreted | example | Save | Load

ARINAS Server (TUXEDO/LOCAL) v15e45+dirty
15e45+dirtybuild #1 (debug), 2025-08-14T09:42:19

INPUT TREE

```
ret{
  /context/online/editor = null;
  /path/check/success = true;
  /content/data = 4;
  /src = "log(global.lang.builtins());";
}
```

ONLINE EDITOR

```
ret/greek/alphabet = string(cmd/path/check) ~ " - alpha";

var/iter = cmd/content/data;

while (var/iter) {
  ret/path[path(ret/path)][`_` ~ var/iter] = true;
  ret[3*cmd/content/data - var/iter*2] = true;
  var/iter--;
}
__EOF__
exec{
  /pattern = cur;
  /source{
    /type/les = 1.0;
    /name = "ONLINE";
    /code = cmd/src;
  }
  /input = null;
};

ret/builtins/raw = global.lang.builtins();
foreach (ret/builtins/raw)
  ret/builtins[] = key.replace("/", ".", "code.", "") ~ "()";
```

OUTPUT TREE

```
ret{
  /greek/alphabet = "success - alpha"
  /path/_4/_3/_2/_1 = true;
  ["4"] = true;
  ["6"] = true;
  ["8"] = true;
  ["10"] = true;
}
```

LOGS: [2025-08-19 00:25:52.6132482] Run time 741 ms, 465 µs, and 3 hnsecs - +

```
LOG START [2025-08-19]
[62E3149E:67A7F342:128 00:25:52.6033808]  LESCODE INFO  [arinas.platform.builtin.code.info.exec:73] INFO in method: /project/editor/content/main.prerender() on Ln 26, Col 3:
var/result [Main page]
    ├─ greek ── alphabet = "success - alpha" (string[15])
    ├─ path ── _4 ── _3 ── _2 ── _1 = true (bool)
    ├─ 4 = true (bool)
    ├─ 6 = true (bool)
    ├─ 8 = true (bool)
    └─ 10 = true (bool)
```


PROJECTS
BACKBONE
PROJECT / LeS editor
PROJECT / Arinas IDE
PROJECT / sample
/editor
/example
MODULES
program | Program modules
content | Content layer
root | Content layer root (/content
main | Main application webpage (/c
editor | Code editor (/content/item
prerender()
portal | Portal modules
Editor
Synchronization
Tools
Save
Operation: content/editor.prerender()
1 ret/render{
2 /renderer/content/editor = null;
3 /control{
4 /context = cmd/context;
5 }
6 /params{
7 /name = cfg/label ?? cfg/name ?? "EDITOR";
8 /sourceCode = cmd/content/code ?? cfg/defaultCode ?? "";
9 /instance = string(path.tail(cmd/context));
10 /editor[tmp/instance]{
11 /line = 1;
12 /ch = 1;
13 /left = 0;
14 /top = 0;
15 }
16 }
17 }
ARINAS Server (tuxedo/local@local) v15e45+dirty
build #1 (debug/devel), 2025-08-14 09:42:19
PROGRAM
program | Interactive LeS code Online Editor (program/root)
content | Content layer (content/root)
app | Application portal (content/portal/default)
template | HTML template
main | Main page (content/main)
template | HTML template
cmdEditor | Input tree editor (content/editor)
codeEditor | Code editor (content/editor)
retEditor | Output tree editor (content/editor)
security | Secutrity layer (/security/root)
render | Rendering layer (/render/root)

Arinas Platform – Live Demo

- Live examples
  - locating a component
  - implementing requirements at various levels
  - instantiation and use in different contexts
- Architectural levels
  - top level
  - layer
  - component
  - element / function
  - configuration

Arinas Platform – LeS language

- An interpreted and extensible programming language for tree manipulation
- A module's operation in Arinas can be either:
  - stateless: f(cur, cmd) = (ret)
  - stateful: f(cur, cmd, env, req, ses) = (ret, req, ses)
- Concise syntax for expressing structure and behaviour

Arinas Platform – LeS language

- LeS language comprises
  - **control** structures: if-else, for, foreach, while, switch, in (loop)
  - tree **context** binding: *src*, *tmp*, *dst* trees, command blocks
  - **value** handling: *assignments*, *expressions*, *builtins*
  - **exception** handling: try, throw, catch, finally
  - **jump** statements: return, break, skip, continue
  - object hierarchy **tests** and traversals: eq, isof, refines, this, super, root, parentof

Arinas Platform - Implementation

- Crucial components
  - DMD + dub
  - Zsh interpreter
  - vibe.d HTTP server
  - vibe-core + all vibe.d dependencies
  - database connectors (optional)
    - PostgreSQL (dpq2), SQLite
- 18.5k effective lines of D code in 62 files
- 32,5k LOC in total

Language	files	blank %	comment %	code
D	62	11.37	31.74	18507

Why D was chosen

- Native compilation – runs fast
- Multi-paradigm approach (assembler to metaprogramming)
- Open source and free software – compiler, libs
- Ample learning resources
- Advanced usability features – UFC, CTFE, RAII
- Crucial ecosystem components – such as vibe.d

Appreciated features of D

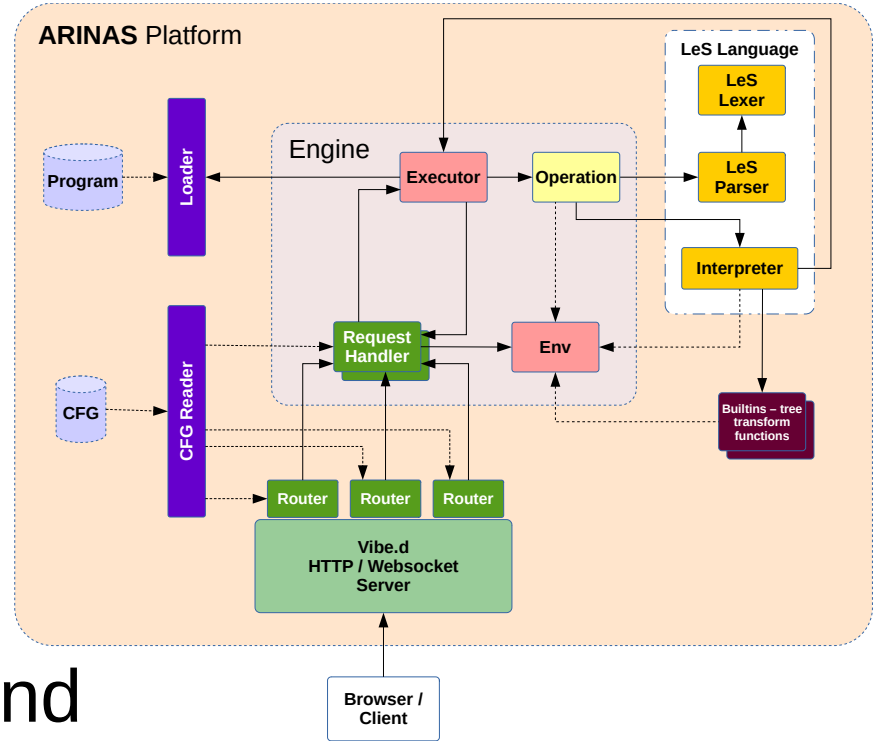
- Generic programming
- Metaprogramming
- Compile-time introspection
- Declarative programming
  - e.g. when writing the lexer

Appreciated features of D

- Linker instructions (e.g. injecting mangled names for a function)
- Sleek, well-structured documentation
- Fast compilation – rapid development cycle
- Built-in boilerplate elimination
- Expressive, C/C++-syntax style

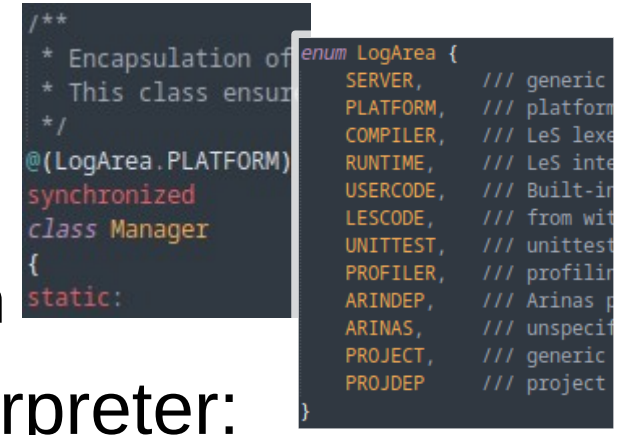
D language – implementation support

- D has been effective for
  - HTTP server – vibe.d
  - runtime configuration – layering of custom, default, and fallback option values
  - LaexTree data structure – templated leaf data types and LaexTreeVirtual node type



D language – implementation support

- D has been effective for
  - segmented logging: log areas defined with UDAs and compile-time introspection
  - LeS language lexer, parser and interpreter: by reusing the D lexer and declaring LeS lexemes and rules
  - additional infrastructure: LeS language *builtins* – templated mixin definitions, timers – event-loop based, database connectors – ddbc, dpq2



D language – lessons learned

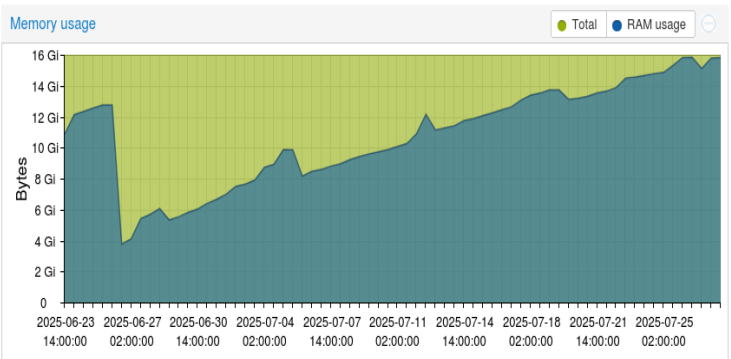
- Strengths:
  - Quick compilation – fast development cycle (performance)
  - Versatile multiparadigm and pragmatic approach (flexibility and freedom)
  - Well-structured documentation

D language – lessons learned

- Improvement suggestions:
  - vibe.d – gradually fragmenting
  - garbage collector efficiency and usability
  - non-intuitive idioms, e.g. scope parameters, @safe escapes with

```
@trusted { doUnsafe(); }()
```

  - lack of official documentation on advanced features and topics



The D community – observations

- Talented people make all the difference – early potential turned into results
  - a spirit of unbiased innovation – real technical solutions
  - a pragmatic approach – rooted in practice
- To use D is to have a lot of fun

Value created with D

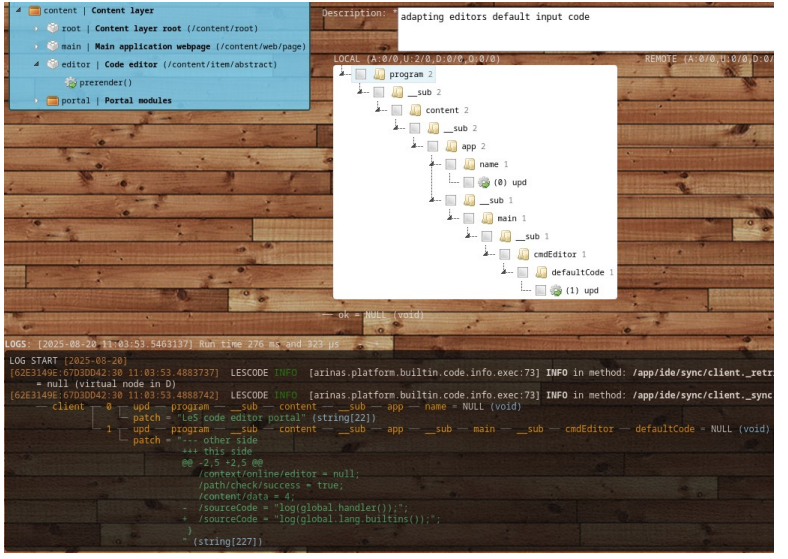
Arinas Platform benefits

- clear visibility of the layered architecture
- clean architectural interfaces - enabling replacement or reimplementation
- top-down design and coding experience
- manageable module upgrading
- controlled refactoring and impact management

Value created with D - cont’d

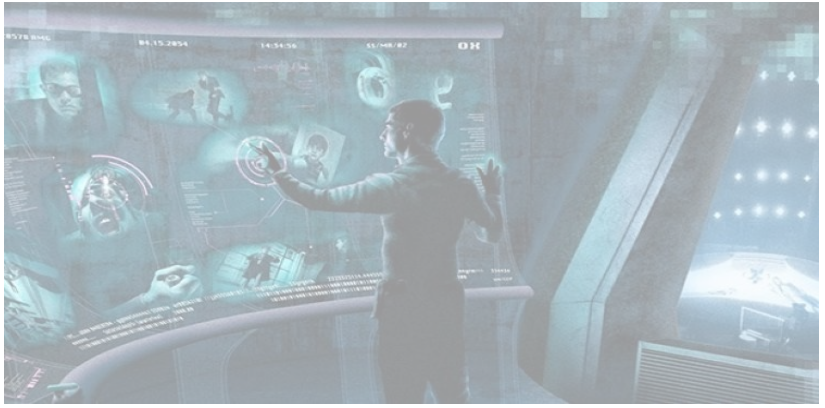
Arinas Platform benefits

- rapid data processing of tree data structures
- top-down design verified by direct execution
  - from experience: ‘when it works, it’s correct’
- context synchronization and versioning at node/attribute level



Arinas Platform – Vision and Future

- Architectural orchestrator – polyglot approach
- Cloud deployments – supporting multitenancy
- AI-assisted development – leveraging pattern matching for suggestions and derivations



Arinas Platform – Vision and Future

- 3D visualization – VR, AR, **holographic projections**
- D language could become a viable alternative in VR development

- Promote D language as a robust foundation for higher-level concepts and architectures

Key Takeaways

- When your goal is to run any complex system with maximum efficiency, the **D language** is the right tool — delivering outstanding computational and system performance.
- When you also want to **build** such systems efficiently — reducing development overhead, streamlining workflows, and accelerating delivery — that’s where **Arinas Platform** comes in.

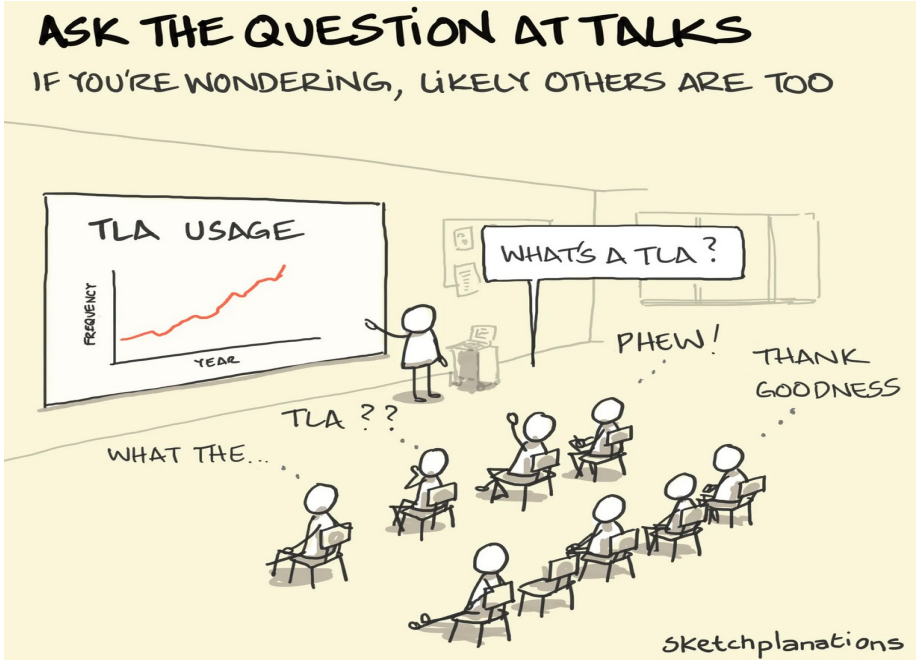
Thank you very much...

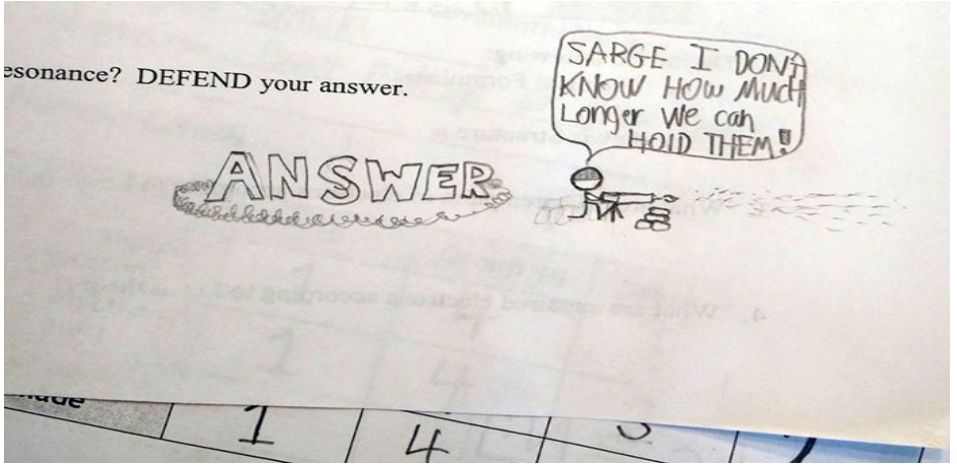
... for the  language

Arinas Platform

A visible architecture made easy with

- Questions?





- Answers?

arinas.org

lescode.arinas.net

bitbucket.digital-orchestra.sk/projects/ARINAS/repos/public