CED-2026 · ECONOMIC DEVELOPMENT AND THE LEGACY OF SIMON KUZNETS
SECTION 4: DIGITAL WORLDS AND SMART TECHNOLOGIES · KHARKIV, 29 OCTOBER 2026

A Conceptual Model for Integrating External Information Services into the Web-Oriented Information System of a Higher Education Institution

An integration layer described in terms of sets and mappings, and a verifiable proposition about adding new providers

Roman Yatsenko · Illia Shyian

R. Yatsenko, PhD (Econ.), Associate Professor, Department of Intelligent Control Systems, scientific supervisor
I. Shyian, PhD student, Department of Intelligent Control Systems

Simon Kuznets Kharkiv National University of Economics

THE PROBLEM

Every external service a university connects to is a separate cost

A university’s information system works with state registries, learning platforms and accounting systems whose contracts the university does not define.

Each connection has its own development and support costs.

Direct links among n systems: at most

$$n(n - 1) / 2$$

Four systems: up to 6 links. Ten: up to 45. This growth, not the complexity of any single link, drives most of the maintenance cost.

WHAT EXISTS

Engineering practice solves the parts, one pattern at a time

Integration patterns [1]

Ways to exchange messages between applications

API gateway [2]

One entry point for the clients of a distributed system

Adapter [3]

Turns a foreign interface into the expected one

Microservices [4]

Conditions for components to stay autonomous

The gap: these sources, and Ukrainian studies [5–7], mostly describe each solution separately. The integration layer as a whole has received little attention so far, and extensibility is usually presented as a general property of an architectural style, without explicit conditions or limits.

AIM OF THE WORK

Build a conceptual model of the integration layer and its formal description, from which a verifiable proposition about extensibility follows.

Extensibility here means one thing: extensibility with respect to the set of providers.

INPUT TO THE MODEL

Six requirements determine what the model contains

1
Single canonical data model

2
Single failure classification and retry policy

3
Authentication as an extension point

4
Uniform abstraction for sequential data retrieval

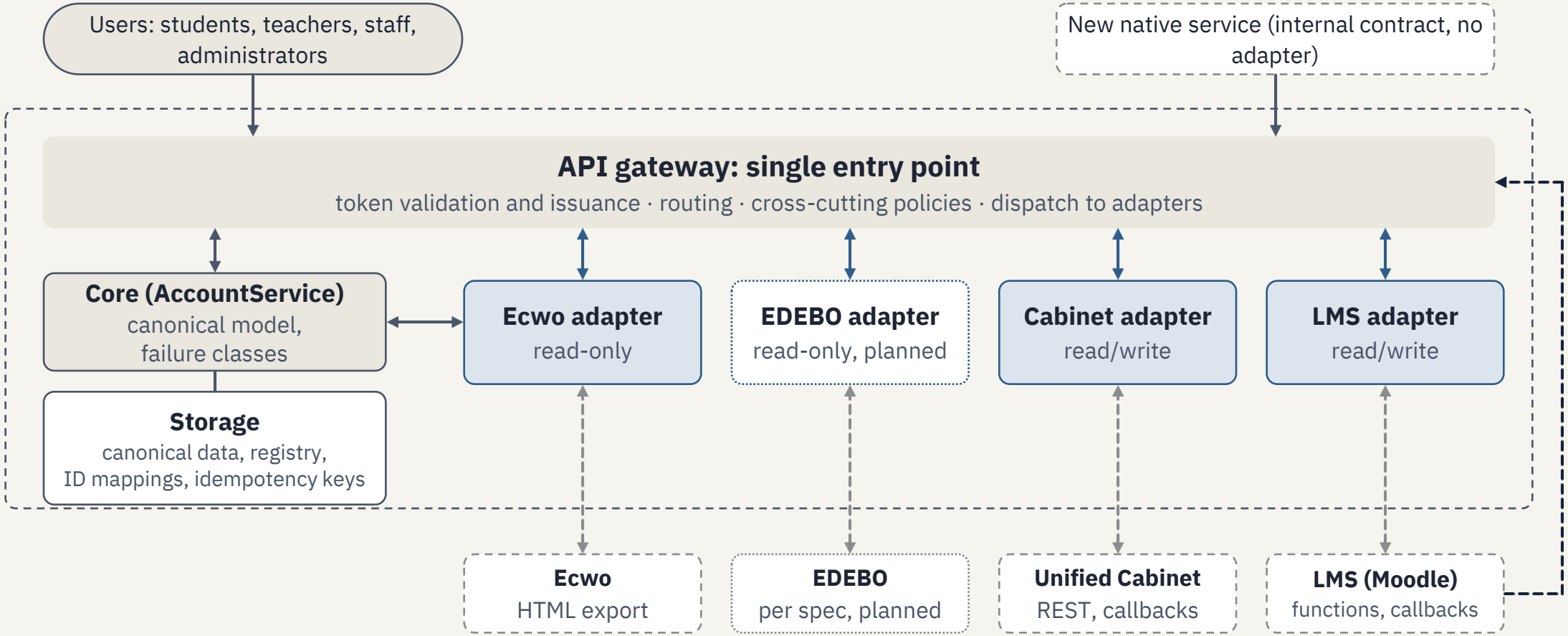
5
Idempotent processing on the consumer side

6
Single internal adapter contract

Derived earlier by the author from observable properties of external APIs. Here they are taken as given.

THE MODEL, CONTAINER LEVEL

Gateway, core, storage and adapters form the platform



FORMAL DESCRIPTION

The model is a four-tuple; none of a provider’s components is defined by the system

$$M = (C, P, A, G)$$

$$C = (O, K, F)$$

C internal contract: unified operations *O*, canonical data model *K*, failure classes *F*

P finite set of external providers, **A** adapters, **G** API gateway

A provider is a tuple: a set of endpoints *E*, a data representation *D*, a set *R* of responses without a usable result, an authentication mechanism and an access mode

None of a provider’s components is defined by the system.

THE ADAPTER

An adapter for provider *p* is a tuple of four mappings

$$a_p = (\varepsilon_p,\ \delta_p,\ \delta_p',\ \varphi_p)$$

Mapping	From → to	What it means
ε	operation O → endpoint E	Partial: its domain is the set of operations the adapter actually implements
δ	canonical K → provider D	Forward conversion of data to the provider's representation
δ′	provider D → canonical K	Reverse conversion back to the canonical model
φ	response R → class F	Defined on all of R: no failure without a defined reaction reaches the core

MAIN RESULT: PROPOSITION

Under four conditions, a new provider leaves the contract, the gateway and existing adapters unchanged

Let *P* be the set of *n* providers, and let a new provider, not in *P*, have needs that fit the existing contract *C*.

1 · Contract only

The core interacts with the integration layer only through *C*

2 · Registry routing

Routing is factored through the provider registry

3 · Internal identity

An external identifier is not a key in any core entity

4 · Known domain

The gateway knows each adapter's domain before the request is executed

$$M' = (C,\ P \cup \{p_{n+1}\},\ A \cup \{a_{n+1}\},\ G)$$

The contract *C*, the gateway *G* and every existing adapter coincide with the original ones.

LIMITS OF THE PROPOSITION

The proposition covers new providers, not a new contract

Covered

Extension of the set of providers: a provider whose needs fit the existing contract *C*.

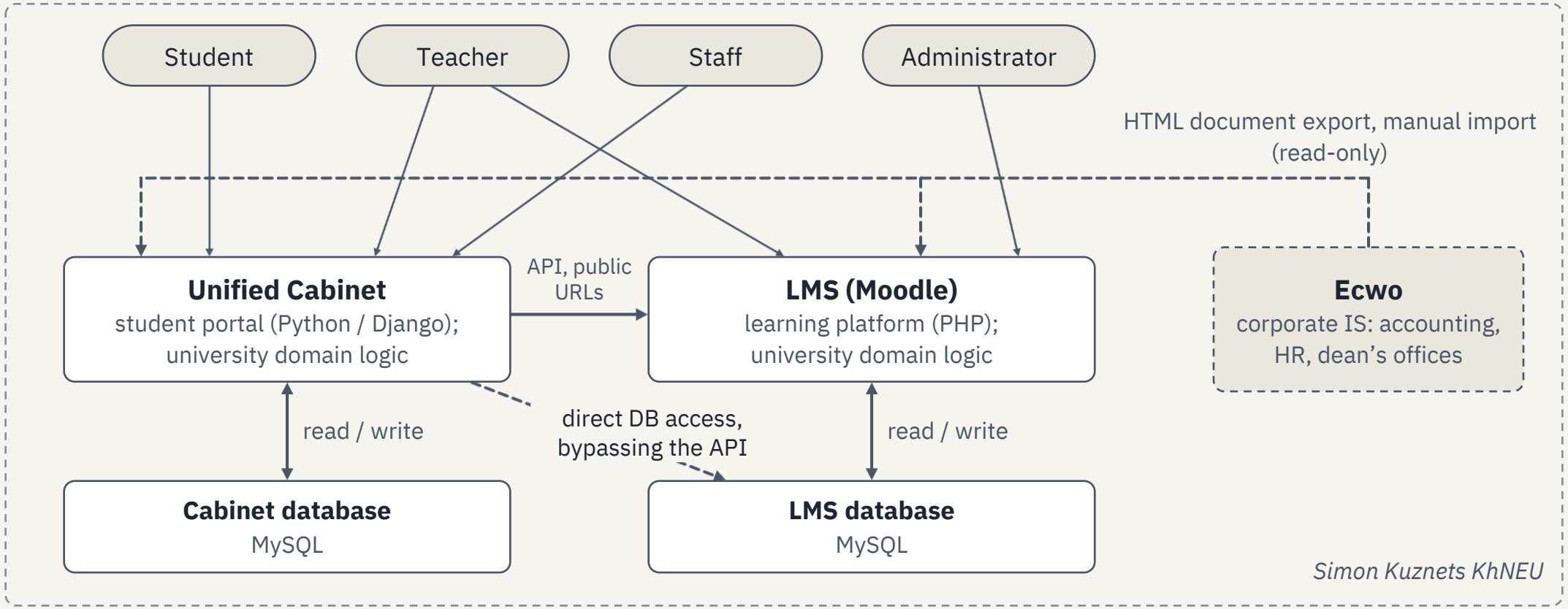
Not covered

A provider that needs a new operation changes the contract, and with it, potentially, the core.

THE UNIVERSITY CASE TODAY

Today: direct calls, manual transfer, API bypass

Solid: programmatic interaction · dashed: manual procedure or bypassing the API · dashed outline: system outside the managed layer. Each system keeps its own list of users.



ECONOMIC CONSEQUENCE

Through the gateway, links grow linearly, not quadratically

Direct links, at most

$$n(n - 1) / 2$$



Each source connects only to the gateway

$$n$$

CONCLUSIONS

What is new: extensibility stated with explicit conditions and an explicit limit

Done

- Conceptual model of the integration layer
- Formal description in terms of sets and mappings
- A verifiable proposition about extensibility with respect to the set of providers

Next

- A method for adaptive interaction of system components with external services, based on a centralised gateway
- Experimental verification of the model

Thank you