ERP/1: Авторизація

Техноробочий проект

на створення та впровадження модуля «Служба ідентифікації та авторизації (IAS)» як засобу інформатизації

Згідно з Постановою КМУ № 205 від 21.02.2025

Формалізація вимог за стандартом ISO/IEC/IEEE 29148:2018

Зміст

Зміст

1 Загальні відомості про засіб інформатизації 2
1.1 Найменування та підстави для розробки . . . 2
1.2 Простежуваність вимог та відповідність нормативній базі . . . 2
1.3 Призначення та цілі створення засобу . . . 2
1.4 План-графік виконання етапів робіт . . . 3

2 Відомості про робочий процес та умови експлуатації 4
2.1 Опис бізнес-процесів (BPMN / FSM) . . . 4
2.1.1 Процес реєстрації пристрою та імпорту OVPN (bpe_ovpn_import.erl) . 4
2.1.2 Процес випуску та сертифікації CMP (bpe_cmp_enrollment.erl) . . . 4
2.1.3 Процес авторизації та оцінки довіри (bpe_trust_evaluation.erl) . . . 4
2.1.4 Процес провіженінгу VPN (bpe_vpn_provisioning.erl) . . . 5
2.2 Опис ролей та прав доступу (ABAC) . . . 5
2.3 Умови експлуатації та системне середовище . . . 5

3 Інформаційне та програмне забезпечення 6
3.1 Інформаційне забезпечення (Схеми та Дані) . . . 6
3.1.1 Визначення структур даних (Erlang Records) . . . 6
3.1.2 Опис довідників та реєстрових сутностей . . . 7
3.1.3 Алгоритми оцінки статусу довіри (Trust Evaluation) . . . 7
3.2 Програмне забезпечення (Реалізація) . . . 8
3.2.1 bpe_ovpn_import.erl (DSL процесу імпорту) . . . 8
3.2.2 bpe_cmp_enrollment.erl (DSL процесу CMP випуску) . . . 8
3.2.3 bpe_trust_evaluation.erl (DSL процесу перевірки довіри) . . . 9
3.2.4 bpe_vpn_provisioning.erl (DSL процесу провіженінгу VPN) . . . 10
3.2.5 ias_validator.erl (Модуль валідації) . . . 10

4 Вимоги до засобу за ISO/IEC/IEEE 29148 12
4.1 Функціональні вимоги . . . 12
4.2 Вимоги до інтерфейсів та дизайну . . . 12
4.3 Вимоги до безпеки та захисту інформації . . . 12
4.4 Вимоги до продуктивності та надійності . . . 12
4.5 Вимоги до логування та аудиту . . . 12
4.6 Адміністративні та юридичні вимоги . . . 13

1. Загальні відомості про засіб інформатизації

1.1. Найменування та підстави для розробки

Найменування засобу інформатизації: Модуль «Служба ідентифікації та авторизації (IAS)» (далі — Модуль IAS / Система IAS) у складі інформаційної системи управління підприємством ERP/1.

Підстави для розробки:

1. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах».
2. Постанова Кабінету Міністрів України від 29.03.2006 № 373 «Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах».
3. Постанова Кабінету Міністрів України від 21.02.2025 № 205 «Про затвердження Порядку використання засобів інформатизації».
4. Угода про асоціацію між Україною та Європейським Союзом (в частині транскордонної взаємодії електронних довірчих послуг).
5. Технічне завдання на побудову Zencrypted інфраструктури безпеки.

1.2. Простежуваність вимог та відповідність нормативній базі

Для забезпечення відповідності Постанові КМУ № 205 від 21.02.2025 та вимогам міжнародного аудиту, проектування Модуля IAS базується на принципі наскрізної простежуваності (traceability) від високорівневих бізнес-політик безпеки до конкретних елементів реалізації.

Усі вимоги, визначені в `ias.tex`, пов'язані з відповідними структурами даних та виконуваними процесами в цьому документі:

- Політика доступу та ABAC: Політики безпеки з `policy.tex` транслюються у функціональні вимоги контролю доступу в `ias.tex` (Розділ 4.1.5) та формалізуються в цьому документі у вигляді коду перевірки атрибутів (`allow/3` в Розділі 2.2).
- Керування життєвим циклом сертифікатів: Вимоги до CMP-випуску транслюються у реєстрову сутність `Certificate` та процес `bpe_cmp_enrollment.erl` (Розділ 3.2.2).
- Провіженінг та узгодження конфігурацій: Вимоги до одностороннього провіженінгу та реконсиляції реалізуються через сутність `ias_vpn_device_state` та процес `bpe_vpn_provisioning.erl` (Розділ 3.2.4).

1.3. Призначення та цілі створення засобу

Модуль IAS розробляється як центральна адміністративно-координаційна служба для керування цифровими ідентичностями, сертифікатами пристроїв, авторизацією користувачів та віддалених точок підключення VPN.

Основні цілі створення:

- Автоматизація життєвого циклу PKI-сертифікатів клієнтських пристроїв через стандартизовані протоколи CMP, EST та CMC.
- Забезпечення повного контролю за конфігураціями VPN-шлюзів через односторонній транзакційний провіженінг.
- Підтримка відмовостійкості та швидкої реконсиляції мережевого стану у разі аварійних перезапусків та збоїв.
- Відповідність вимогам міжнародного аудиту щодо захисту персональних даних та контролю доступів (ABAC).

1.4. План-графік виконання етапів робіт

Розробка та впровадження Модуля IAS поділяється на такі етапи:

1. Етап 1: Аналітичне моделювання та проектування (розробка схем даних, визначення графів об'єктів та написання BPE-процесів) — 4 тижні.
2. Етап 2: Реалізація сховища даних KVS та бізнес-логіки (реалізація Mnesia-моделей та BPE DSL рушія) — 6 тижнів.
3. Етап 3: Інтеграція з CA та VPN (реалізація CMP-клієнта, Erlang RPC провіженінгу та reconciliation циклів) — 6 тижнів.
4. Етап 4: Розробка веб-інтерфейсу N2O/NITRO (створення адміністративної консолі та wizard-інтерфейсів) — 4 тижні.
5. Етап 5: Державна експертиза та сертифікація КСЗІ (отримання експертного висновку) — 8 тижнів.

2. Відомості про робочий процес та умови експлуатації

2.1. Опис бізнес-процесів (BPMN / FSM)

Всі процеси керування доступом формалізуються у вигляді кінцевих автоматів (FSM) та виконуються рушієм процесів BPE.

2.1.1. Процес реєстрації пристрою та імпорту OVPN (bpe_ovpn_import.erl)

Цей процес відповідає за початкове імпортування раніше створених OpenVPN конфігурацій та створення відповідних реєстрових сутностей в IAS.

1. StartImport: Початок процесу оператором безпеки.
2. ParseConfig: Сервісна задача, яка парсить .ovpn та створює структуру вилучених параметрів.
3. VerifyCertificates: Перевірка валідності імпортованого сертифіката (підпис, термін дії, ланцюжок довіри).
4. ApproveImportPlan: Користувацька задача затвердження згенерованого плану імпорту оператором.
5. CreateGraphObjects: Сервісна задача, яка записує об'єкти Device, Certificate, VPN Service та зв'язки між ними в KVS.
6. ImportCompleted: Завершення процесу.

2.1.2. Процес випуску та сертифікації CMP (bpe_cmp_enrollment.erl)

Процес забезпечує автоматизоване отримання сертифіката від CA за допомогою CSR, згенерованого безпосередньо клієнтським пристроєм.

1. StartEnrollment: Ініціація пристроєм запиту на випуск.
2. GenerateKeyCSRPlan: Створення в IAS плану генерації ключів та CSR на пристрої.
3. AwaitCSRUpload: Очікування завантаження файлу CSR з пристрою.
4. ValidateCSR: Перевірка підпису та параметрів CSR.
5. SubmitToCMP: Відправка CSR до CA через протокол CMP.
6. VerifyCertificate: Отримання випущеного сертифіката, перевірка публічного ключа проти CSR, запис у сховище та оновлення статусу пристрою.
7. EnrollmentCompleted: Завершення випуску.

2.1.3. Процес авторизації та оцінки довіри (bpe_trust_evaluation.erl)

Процес динамічної оцінки права пристрою чи користувача на підключення.

1. StartEvaluation: Виклик процесу під час handshake пристрою або планової перевірки.
2. CheckExpiry: Перевірка дати закінчення дії сертифіката.
3. CheckCRL: Запит та перевірка серійного номера за списками відкликання (CRL/OCSP).
4. EvaluateABACRules: Оцінка контекстних правил (роль, ЄДРПОУ, час, робоче місце).
5. UpdateTrustStatus: Запис оновленого статусу довіри пристрою (Ready або Blocked).

6. EvaluationCompleted: Завершення оцінки.

2.1.4. Процес провіженінгу VPN (bpe_vpn_provisioning.erl)

Процес транзакційної доставки налаштувань на шлюзи VPN та контролю узгодження.

1. StartProvisioning: Ініціація оновлення конфігурації.
2. BuildCommand: Генерація канонічної команди провіженінгу з інкрементним номером ревізії.
3. DeliverToVPN: Виклик віддаленої процедури RPC на шлюзі VPN.
4. AwaitCompletionNotice: Очікування повідомлення від vpn_event_bus про успішне застосування змін на шлюзі.
5. VerifyDataplane: Перевірка проходження тестового пакету через зашифрований тунель.
6. ProvisioningCompleted: Завершення провіженінгу.

2.2. Опис ролей та прав доступу (ABAC)

Доступ до адміністративних функцій IAS обмежується на базі ABAC. Приклад правила авторизації:

```
allow(Operator, Action, Object) ->
    OperatorRole = maps:get(role, Operator),
    OperatorEDRPOU = maps:get(edrpou, Operator),
    ObjectSensitivity = maps:get(sensitivity_level, Object),

    IsSecurityAdmin = (OperatorRole == security_admin),
    MatchesEDRPOU = (OperatorEDRPOU == maps:get(edrpou, Object, undefined)),

    case {IsSecurityAdmin, ObjectSensitivity} of
        {true, _} -> true;
        {false, confidential} -> MatchesEDRPOU;
        {false, public} -> true;
        _ -> false
    end.
```

2.3. Умови експлуатації та системне середовище

Система IAS повинна розгортатися у кластерному середовищі:

- Операційна система: Linux (Ubuntu Server LTS, Rocky Linux) або macOS (для розробки).
- Платформа: Erlang/OTP версії 28+.
- Служби взаємодії: Розподілені Erlang-ноди, з'єднані за допомогою epmd та TLS mesh.
- Вимоги до СУБД: Mnesia, інтегрована через інтерфейс KVS.

3. Інформаційне та програмне забезпечення

3.1. Інформаційне забезпечення (Схеми та Дані)

3.1.1. Визначення структур даних (Erlang Records)

Всі ключові сутності описуються у вигляді Erlang рекорді:

```
-record(ias_domain_object, {
    key,                        % {Kind, ObjectId}
    schema_version = 1,         % Версія схеми даних
    kind,                       % user | device | certificate | policy
    object_id,                  % Унікальний UUID
    payload = #{},              % Мапа з атрибутами об'єкта
    revision = 1,               % Номер поточної ревізії
    created_at = 0,             % Timestamp створення
    updated_at = 0              % Timestamp оновлення
}).

-record(ias_certificate_material_record, {
    key,                        % {SubjectKind, SubjectId}
    schema_version = 1,
    subject_kind,               % user | device | ca
    subject_id,                 % UUID суб'єкта
    material_type,              % public_cert | cert_chain | signature
    encoding = pem,             % pem | der
    source,                     % cmp | est | manual
    fingerprint_sha256,         % SHA256 відбиток сертифіката
    body_envelope = #{},        % Мапа, що містить PEM-тіло
    revision = 1,
    created_at = 0,
    updated_at = 0,
    expires_at = undefined      % Термін дії сертифіката
}).

-record(ias_vpn_device_state, {
    device_id,                  % UUID пристрою (PK)
    schema_version = 2,
    revision = 0,               % Актуальна ревізія на шлюзі
    command_digest = undefined,% Хеш останньої відправленої команди
    canonical_command = #{},    % Тіло сформованої команди
    binding = #{},              % Параметри мережевої прив'язки
    lifecycle_state = unbound, % unbound | pending | active | degraded
    last_decommission = undefined,
    decommission_history = [],
    decommissioned_at = undefined,
    updated_at = 0
}).

-record(ias_csr_enrollment_record, {
    csr_fingerprint,            % Хеш CSR (PK)
    schema_version = 1,
```

```
    status = submitted,        % submitted | processing | signed | rejected
    retryable = false,         % Чи можна повторити спробу
    payload = #{},             % Мапа з параметрами запиту та CSR PEM
    revision = 1,
    created_at = 0,
    updated_at = 0
}).

-record(ias_vpn_reconciliation_incident, {
    device_id,                 % UUID пристрою
    schema_version = 1,
    kind,                      % orphan | stale_revision | fingerprint_mismatch
    reason,                    % Текстовий опис розбіжності
    token,                     % Унікальний токен інциденту
    status = open,             % open | acknowledged | resolved
    snapshot = #{},            % Порівняльний знімок станів IAS та VPN
    first_seen = 0,
    last_seen = 0,
    occurrences = 1,
    acknowledged_by = undefined,
    acknowledged_note = undefined,
    acknowledged_at = undefined,
    resolved_by = undefined,
    resolved_note = undefined,
    resolved_at = undefined,
    updated_at = 0
}).
```

3.1.2. Опис довідників та реєстрових сутностей

Опис полів реєстрових сутностей (таких як `User`, `Device`, `Certificate`) відповідає таксономії, зазначеній у Технічних вимогах. Всі записи зберігаються у таблиці Mnesia `kvs`, яка містить записи типу `ias_domain_object` та інші допоміжні рекорди.

3.1.3. Алгоритми оцінки статусу довіри (Trust Evaluation)

Приклад визначення статусу пристрою на основі його криптографічного стану:

```
evaluate_device_trust(DeviceState, CertRecord, CurrentTime) ->
    case CertRecord#ias_certificate_material_record.expires_at of
        Expires when CurrentTime > Expires ->
            {blocked, expired};
        _ ->
            case check_crl_status(CertRecord#ias_certificate_material_record.fingerprint_sha256) o
                revoked -> {blocked, revoked};
                ok ->
                    case DeviceState#ias_vpn_device_state.lifecycle_state of
                        degraded -> {degraded, connection_timeout};
                        active -> {ready, ok};
                        unbound -> {incomplete, no_binding}
                    end
            end
    end.
```

3.2. Програмне забезпечення (Реалізація)

3.2.1. bpe_ovpn_import.erl (DSL процесу імпорту)

```
-module(bpe_ovpn_import).
-include("bpe.hrl").
-compile(export_all).

def() ->
    #process{
        name = 'OVPN Profile Import',
        beginEvent = 'StartImport',
        endEvent = 'ImportCompleted',
        tasks = [
            #beginEvent{name='StartImport'},
            #serviceTask{name='ParseConfig', module=?MODULE},
            #serviceTask{name='VerifyCertificates', module=?MODULE},
            #userTask{name='ApproveImportPlan', module=?MODULE},
            #serviceTask{name='CreateGraphObjects', module=?MODULE},
            #endEvent{name='ImportCompleted'}
        ],
        flows = [
            #sequenceFlow{name='1', source='StartImport', target='ParseConfig'},
            #sequenceFlow{name='2', source='ParseConfig', target='VerifyCertificates'},
            #sequenceFlow{name='3', source='VerifyCertificates',
                          target='ApproveImportPlan', condition="certs_ok"},
            #sequenceFlow{name='4', source='VerifyCertificates',
                          target='StartImport', condition="certs_invalid"},
            #sequenceFlow{name='5', source='ApproveImportPlan', target='CreateGraphObjects'},
            #sequenceFlow{name='6', source='CreateGraphObjects', target='ImportCompleted'}
        ],
        roles = [security_admin, operator]
    }.

action({serviceTask, 'ParseConfig'}, Proc) ->
    % Логіка парсингу OVPN файлу
    {reply, Proc, "certs_ok"};
action({serviceTask, 'CreateGraphObjects'}, Proc) ->
    % Створення сутностей в KVS
    {reply, Proc};
action(_, Proc) -> {reply, Proc}.
```

3.2.2. bpe_cmp_enrollment.erl (DSL процесу CMP випуску)

```
-module(bpe_cmp_enrollment).
-include("bpe.hrl").
-compile(export_all).

def() ->
    #process{
        name = 'CA CMP Enrollment',
        beginEvent = 'StartEnrollment',
        endEvent = 'EnrollmentCompleted',
        tasks = [
            #beginEvent{name='StartEnrollment'},
            #serviceTask{name='GenerateKeyCSRPlan', module=?MODULE},
```

```
            #userTask{name='AwaitCSRUpload', module=?MODULE},
            #serviceTask{name='ValidateCSR', module=?MODULE},
            #serviceTask{name='SubmitToCMP', module=?MODULE},
            #serviceTask{name='VerifyCertificate', module=?MODULE},
            #endEvent{name='EnrollmentCompleted'}
        ],
        flows = [
            #sequenceFlow{name='1', source='StartEnrollment', target='GenerateKeyCSRPlan'},
            #sequenceFlow{name='2', source='GenerateKeyCSRPlan', target='AwaitCSRUpload'},
            #sequenceFlow{name='3', source='AwaitCSRUpload', target='ValidateCSR'},
            #sequenceFlow{name='4', source='ValidateCSR', target='SubmitToCMP', condition="csr_val
            #sequenceFlow{name='5', source='ValidateCSR', target='AwaitCSRUpload', condition="csr_
            #sequenceFlow{name='6', source='SubmitToCMP', target='VerifyCertificate', condition="e
            #sequenceFlow{name='7', source='SubmitToCMP', target='StartEnrollment', condition="enr
            #sequenceFlow{name='8', source='VerifyCertificate', target='EnrollmentCompleted'}
        ],
        roles = [operator, device]
    }.

action({serviceTask, 'ValidateCSR'}, Proc) ->
    {reply, Proc, "csr_valid"};
action({serviceTask, 'SubmitToCMP'}, Proc) ->
    {reply, Proc, "enrollment_ok"};
action(_, Proc) -> {reply, Proc}.
```

3.2.3. bpe_trust_evaluation.erl (DSL процесу перевірки довіри)

```
-module(bpe_trust_evaluation).
-include("bpe.hrl").
-compile(export_all).

def() ->
    #process{
        name = 'Device Trust Evaluation',
        beginEvent = 'StartEvaluation',
        endEvent = 'EvaluationCompleted',
        tasks = [
            #beginEvent{name='StartEvaluation'},
            #serviceTask{name='CheckExpiry', module=?MODULE},
            #serviceTask{name='CheckCRL', module=?MODULE},
            #serviceTask{name='EvaluateABACRules', module=?MODULE},
            #serviceTask{name='UpdateTrustStatus', module=?MODULE},
            #endEvent{name='EvaluationCompleted'}
        ],
        flows = [
            #sequenceFlow{name='1', source='StartEvaluation', target='CheckExpiry'},
            #sequenceFlow{name='2', source='CheckExpiry', target='CheckCRL', condition="not_expire
            #sequenceFlow{name='3', source='CheckExpiry', target='UpdateTrustStatus', condition="e
            #sequenceFlow{name='4', source='CheckCRL', target='EvaluateABACRules', condition="not_
            #sequenceFlow{name='5', source='CheckCRL', target='UpdateTrustStatus', condition="revo
            #sequenceFlow{name='6', source='EvaluateABACRules', target='UpdateTrustStatus'},
            #sequenceFlow{name='7', source='UpdateTrustStatus', target='EvaluationCompleted'}
        ],
        roles = [security_admin, system]
    }.
```

```
action({serviceTask, 'CheckExpiry'}, Proc) ->
    {reply, Proc, "not_expired"};
action({serviceTask, 'CheckCRL'}, Proc) ->
    {reply, Proc, "not_revoked"};
action(_, Proc) -> {reply, Proc}.
```

3.2.4. bpe_vpn_provisioning.erl (DSL процесу провіженінгу VPN)

```
-module(bpe_vpn_provisioning).
-include("bpe.hrl").
-compile(export_all).

def() ->
    #process{
        name = 'VPN Command Provisioning',
        beginEvent = 'StartProvisioning',
        endEvent = 'ProvisioningCompleted',
        tasks = [
            #beginEvent{name='StartProvisioning'},
            #serviceTask{name='BuildCommand', module=?MODULE},
            #serviceTask{name='DeliverToVPN', module=?MODULE},
            #userTask{name='AwaitCompletionNotice', module=?MODULE},
            #serviceTask{name='VerifyDataplane', module=?MODULE},
            #endEvent{name='ProvisioningCompleted'}
        ],
        flows = [
            #sequenceFlow{name='1', source='StartProvisioning', target='BuildCommand'},
            #sequenceFlow{name='2', source='BuildCommand', target='DeliverToVPN'},
            #sequenceFlow{name='3', source='DeliverToVPN', target='AwaitCompletionNotice', conditi
            #sequenceFlow{name='4', source='DeliverToVPN', target='StartProvisioning', condition="
            #sequenceFlow{name='5', source='AwaitCompletionNotice', target='VerifyDataplane'},
            #sequenceFlow{name='6', source='VerifyDataplane', target='ProvisioningCompleted'}
        ],
        roles = [operator, system]
    }.

action({serviceTask, 'BuildCommand'}, Proc) ->
    {reply, Proc};
action({serviceTask, 'DeliverToVPN'}, Proc) ->
    {reply, Proc, "delivery_ok"};
action(_, Proc) -> {reply, Proc}.
```

3.2.5. ias_validator.erl (Модуль валідації)

```
-module(ias_validator).
-export([validate_csr/1, validate_chain/2, verify_fingerprint/2]).

validate_csr(CSR_PEM) ->
    % Виклик OpenSSL або вбудованого ASN.1 розшифровувача для перевірки підпису CSR
    case public_key:pem_decode(CSR_PEM) of
        [_] -> ok;
        _   -> {error, invalid_csr_format}
    end.

validate_chain(Cert_PEM, CA_Chain_PEM) ->
    % Криптографічна перевірка ланцюжка сертифікатів CA
```

```
    ok.

verify_fingerprint(Cert_PEM, SHA256) ->
    % Порівняння відбитку SHA-256
    Calculated = crypto:hash(sha256, Cert_PEM),
    case Calculated of
        SHA256 -> ok;
        _      -> {error, fingerprint_mismatch}
    end.
```

4. Вимоги до засобу за ISO/IEC/IEEE 29148

4.1. Функціональні вимоги

Система повинна забезпечувати повнофункціональне керування ідентичностями, валідацію сертифікатів, оновлення конфігурацій шлюзів VPN без втрати з'єднань користувачів та виявлення невідповідностей стану мережі.

4.2. Вимоги до інтерфейсів та дизайну

- Веб-інтерфейс адміністративної консолі повинен базуватися на реактивному фреймворку N2O/NITRO.
- Відповідність вимогам WCAG 2.1 рівню AA для доступності інтерфейсів.
- Візуальний дизайн повинен наслідувати фірмовий стиль Zencrypted з підтримкою Sleek Dark Mode.

4.3. Вимоги до безпеки та захисту інформації

- Шифрування каналів зв'язку за допомогою TLS v1.3.
- Підтримка КЕП CAdES-X Long та алгоритму ДСТУ 4145-2002.
- Ізоляція ключів: приватний ключ ніколи не записується і не передається в IAS, залишаючись виключно на кінцевому пристрої.
- Категорична відмова від використання JSON Web Tokens (JWT) та OpenID Connect (OIDC) як основних засобів автентифікації. Замість цього застосовується криптоцентрична архітектура на базі mTLS X.509 та локальної генерації CSR на пристроях відповідно до [1, 2].
- Атестація комплексу за вимогами ДССЗІ України на рівень захисту Г3.

4.4. Вимоги до продуктивності та надійності

- Час аптайму системи не менше 99.99% (High Availability).
- Час відновлення (RTO) після повного перезапуску ноди не більше 5 секунд завдяки KVS/Mnesia rehydration.
- Здатність обробляти до 1000 запитів авторизації на секунду на одну ноду.

4.5. Вимоги до логування та аудиту

- Всі операції провіженінгу, випуску та зміни статусів довіри повинні підписуватися цифровим підписом ноди та записуватися в стійкий лог аудиту.
- Заборона модифікації або видалення записів журналу аудиту (immutable audit log).

4.6. Адміністративні та юридичні вимоги

- Всі компоненти поставляються з ліцензіями відкритого вихідного коду (MIT/Apache 2.0).
- Модуль IAS повинен інтегруватися з наявною інфраструктурою підприємства без потреби закупівлі додаткового платного ПЗ.

Література

[1] Технічні вимоги: Про неприпустимість використання JWT токенів як самостійного механізму в державних системах, 2026. `https://5ht.co/jwt.pdf`

[2] Технічні вимоги: Щодо неприпустимості використання OpenID Connect токенів при наявності eIDAS/EUDI у вимогах, 2026. `https://5ht.co/openid.pdf`