ПЕРША ДЕРЖАВНА СИСТЕМА

Формальна модель та програмна архітектура
функціонального верифікованого мовного забезпечення
для побудови інфраструктурних процесінгових систем
орієнтованих на державну модель управління:
процесами для проведення зовнішнього аудиту,
різними видами розподілених сховищ,
телекомунікаційними та реєстровими фреймворками,
інтернет-утворюючими сервісами зокрема та для
автоматизації захищених автономних офісів і
державних підприємств України у цілому.

Навчальний посібник курсу «Інформаційні системи»

Максим Сохацький
18 лютого 2024, Київ, Україна

УДК 002
УДК 004.4, 004.6, 004.9

Присвячується всім державним
службовцям України

Система управління державними підприємствами ERP/1 визначає формальну специфікацію та її імплементацію для сучасних оптимізованих підприємств які вимагають сучасних засобів контролю операцій та цілістності даних.

Телекомунікаційна платформа Erlang/OTP від Ericsson успішно застосовується в індустрії мобільними операторами понад 30 років, а її віртуальна машина досі вважається однією з найкращих в галузі. Системи ERP на її базі також уже не один рік використовується у банківській сфері, процесінгу транзакцій, розподілених системах повідомлень, в IoT секторі. Ви можете переглянути демо модулі системи ERP/1 в нашому захищеному середовищі зі своїм центром випуску ECC X.509 сертифікатів. У цій книзі ви знайдете класичну авторську монографію на тему архітектури та імплементації такої системи, побудованої на міжнародних та державних стандартах України:

RFC: 7363, 6350, 4180, 5126, 5652, 8567, 9006, 9011, 9019, 9159, 9100, 8323, 7815, 7228, 6455, 8927, 8259, 4627, 7493, 7159, 4227, 3288, 6025, 5911, 4120, 4122, 7363, 6537, 6940, 7890, 2251-2256, 6960, 5280, 1034-1035, 4033-4035.

ISO: 19510, 19514, 42010, 18033, 14888, 10118, 10116, 15946, 29146, 9075, 27001, 19464, 20922, 21823, 27402, 30161, 30165, 20452, 42010, 19501, 19505, 8824-8825.

NIST: 800-162.

ДСТУ: 28147, 15946, 9798, 4145, 319-422, 319-122.

Постійне посилання твору: https://axiosis.top/sep/
Видавець: Державний науково-дослідний інститут МВС України

ISBN — 978-617-8027-23-0

Підготовлено до друку на Подолі, м. Київ.

Зміст

1 Вступ 1

2 Національна програма інформатизації 3
2.1 Загальні принципи . . . 3
2.2 Модель . . . 4
2.3 Структурне ядро . . . 4
2.4 Протокол державної інституційної транфсормації . . . 5

3 Органи виконавчої влади 7
3.1 Міністерство науки і освіти . . . 7
3.1.1 Структурні підрозділи . . . 7
3.2 Міністерство охорони здоров'я . . . 8
3.3 Міністерство внутрішніх справ . . . 8
3.4 Міністерство закордонних справ . . . 8
3.5 Міністерство оборони . . . 9
3.5.1 Мотивація . . . 9
3.5.2 Модель . . . 10
3.5.3 Принципи . . . 11
3.5.4 Структурні підрозділи . . . 11
3.5.5 Результати . . . 13
3.5.6 Бібліографія . . . 14
3.6 Міністерство юстиції . . . 15
3.7 Міністерство фінансів . . . 15
3.8 Міністерство економіки . . . 15
3.9 Міністерство енергетики . . . 15
3.10 Міністерство соціальної політики . . . 15
3.11 Міністерство регіональної політики . . . 15
3.12 Міністерство цифрової трансформації . . . 15
3.13 Міністерство молоді та спорту . . . 15
3.14 Міністерство природних ресурсів . . . 15
3.15 Міністерство аграрної політики . . . 15

4 Специфікація та сертифікація 17
4.1 Законодавча база 17
4.1.1 Загальні положення 17
4.1.2 Базова версія «MIA: Документообіг» 18
4.1.3 Розширення та додаткові модулі 19
4.2 Класифікація вимог 21
4.2.1 Вимоги до інтерфейсу користувача 21
4.2.2 Вимоги до адміністрування системи 21
4.2.3 Вимоги типових ділопроцесів системи 21
4.2.4 Вимоги процесінгової системи 21
4.2.5 Вимоги інтеграції з зовнішніми системами . . . 21
4.2.6 Вимоги до розподіленої роботи 21
4.2.7 Вимоги до комплексу засобів захисту (КЗЗ) . . 21
4.2.8 Технічні вимоги до зберігання даних 21
4.3 Відповідність міжнародним стандартам 21
4.3.1 Стандарти RFC 21
4.3.2 Стандарти ISO 21
4.3.3 Національні стандарти ДСТУ та NIST 21
4.4 Засоби захисту та ступені гарантії безпеки 21
4.4.1 Мануальна наочна верифікація 21
4.4.2 Інтеграційне тестування 21
4.4.3 Математична верифікація 21

5 Державна система 23
5.1 Юридично-документальний рівень 23
5.2 Обліково-реєстровий рівень 24
5.3 Зв'язність людей та пристроїв 24
5.4 Телекомунікаційна платформа 25
5.5 Схема та метаінформація 25
5.6 Безпека інтернету та інфраструктури 25

6 Юридично-документальний рівень 27
6.1 Вступ . . . 27
6.2 Модулі підприємства . . . 27
6.3 Управління ресурсами . . . 28
6.4 Архітектура CRM системи . . . 29
6.4.1 Сторінки . . . 29
6.4.2 Комболукап . . . 29
6.4.3 Сервіси . . . 29
6.4.4 CEB OBB . . . 29
6.4.5 Шаблони . . . 29
6.4.6 Дерева . . . 29
6.4.7 Процеси . . . 30
6.4.8 Елементи . . . 34
6.4.9 Редактори . . . 38
6.4.10 Конструктор . . . 40

7 Обліково-реєстраційний рівень 41
7.1 Вступ . . . 41
7.1.1 Види реєстрів . . . 41
7.1.2 Функціональні можливості . . . 41
7.2 Модулі підприємства . . . 42
7.3 Архітектура CART системи . . . 42

8 Технологічний рівень зв'язності людей та пристроїв 43
8.1 Вступ . . . 43
8.2 Виробничий процес . . . 44
8.3 Системи сховищ даних . . . 44
8.3.1 Реляційні бази даних . . . 44
8.3.2 Бази даних з єдиним простором ключів . . . 44
8.3.3 Шини комунікації та брокери повідомлень . . . 44
8.3.4 Розміщені в пам'яті гарячі дані . . . 44
8.4 Обчислювальні ресурси . . . 45
8.4.1 Накопичувальні ресурси . . . 46
8.5 Типові специфікації . . . 46
8.6 Середовище . . . 47
8.6.1 Бібліотеки . . . 48
8.6.2 Приклади . . . 48
8.7 Бінарні протоколи та мови їх опису . . . 49
8.7.1 Мова опису протоколів ASN.1 . . . 49
8.7.2 Мова опису протоколів Protobuf/GRPC . . . 49
8.7.3 Мова опису протоколів SOAP/XSD/XML . . . 49
8.7.4 Мова опису протоколів N2O.DEV RPC . . . 49
8.8 Формати передачі даних . . . 49
8.8.1 Формати передачі даних ETF/BERT . . . 49
8.8.2 Текстовий формат з метаописом JSON/JTD . . . 49
8.8.3 Колоночний формат з метаописом CSV/CSM . . . 49
8.9 Розробка Інтернет додатків . . . 50
8.9.1 Erlang та сучасний веб . . . 50
8.9.2 DSL vs Шаблони . . . 50
8.9.3 Історія . . . 51
8.9.4 Інтерфейс NITRO . . . 51
8.9.5 Сховище KVS . . . 51
8.9.6 Логіка BPMN . . . 51
8.9.7 Додатки MQTT та WebSocket . . . 51

9 Схема даних, типи, валідація та генерація 53
9.1 Графічні мови представлення метаінформації UML . . . 53
9.2 Алгебраїчні мови та System F . . . 53
9.3 Моделі процесів . . . 53
9.4 Верифікація типів . . . 53
9.5 Генерація SDK та конекторів . . . 53
9.6 Базова схема підприємства ERP/1 . . . 53

10 Інфраструктурний рівень безпеки інтернету 55
10.1 Електронний підпис і цифрова печатка 55
10.1.1 Приклад використання 57
10.2 Криптографічні інформаційні повідомлення 58
10.2.1 Elixir.CMS.ebin 58
10.2.2 CMS-KARI-ECC 59
10.2.3 CMS-KEKRI-KEK 60
10.2.4 CMS-KTRI-RSA 61
10.2.5 KDF 61
10.2.6 AES-KW 62
10.2.7 AES-256 63
10.3 Імплементація CMP сервера у складі АЦСК 64
10.3.1 CSR 65
10.3.2 CMS 67
10.4 Центри сертифікації CA, АЦСК, ЦЗО та ОЗО 72
10.5 Безпечна система доменних імен DNSSEC 72
10.6 Система директорії підприємства LDAP 72
10.7 Протокол розмежування доступу ABAC 72

11 Апробація 73

12 Висновки 75

Список використаних джерел 77

Розділ 1

Вступ

Цей посібник описує формальну модель та програмну архітектуру функціонального верифікованого мовного забезпечення для побудови інфраструктурних процесінгових систем орієнтованих на державну модель управління: процесами для проведення зовнішнього аудиту, різними видами розподілених сховищ, телекомунікаційними та реєстровими фреймворками, інтернет-утворюючими сервісами зокрема та для автоматизації захищених автономних офісів і державних підприємств України у цілому.

Система управління державними підприємствами ERP/1 представлена у посібнику є не тільки ідіоматичним прикладом побудови інформаційних державних та комерційних систем у цілому, але і визначає формальну специфікацію та її імплементацію (з багатьма національними впровадженнями) для сучасних оптимізованих підприємств які вимагають сучасних засобів контролю операцій та цілістності даних.

Телекомунікаційна платформа Erlang/OTP від Ericsson успішно застосовується в індустрії мобільними операторами понад 30 років, а її віртуальна машина досі вважається однією з найкращих в галузі. Системи управління підприємствами та інші інформаційні системи на її базі також уже не один рік використовується у банківській сфері, процесінгу транзакцій, розподілених системах повідомлень, в IoT секторі. Ви можете переглянути демо модулі системи ERP/1 в нашому захищеному середовищі зі своїм центром випуску ECC X.509 сертифікатів. У цій книзі ви знайдете перелік модулів системи та основні сутності схеми.

Універсальна платформа для створення та забезпечення функціонування інформаційних реєстрів баз (банків) даних різних масштабів: від базових міжсистемних довідників та класифікаторів, до високонавантажених корпоративних, місцевих та державних ресурсів. Цей посібник буде корисний всім, хто хоче зрозуміти які інформаційні системи застосовуються і державному і комерційному секторах.

Скорочення

ЄРЗ (Єдиний реєстр зброї НПУ), СУСЗЦЗ (Система управління силами та засобами цивільного захисту ДСНС), ЄІС (Єдина інформаційна система МВС), ЕСОЗ (Електронна система охорони здоров'я МОЗ), ФП МТРЗ (Функціональна підсистема матеріально-технічного та ресурсного забезпечення МВС), НГУ (Національна гвардія України), ДСНС (Державна служба з надзвичайних ситуацій МВС), ГСЦ (Головний сервісний центр МВС).

Розділ 2

Національна програма інформатизації

Мета Національної програми інформатизації (НПІ) — створення цифрового простору як проміжний птап розвитку інформаційного суспільства, прозорого правового середовища, яке захищене і базується на міжнародних стандртах та забезпечує інформаційні потреб та реалізацує права і свободи громадян на основі своєчасної, достовірної та повної інформації, підвищення ефективності державного управління.

Національна програма інформатизації (НПІ) веде свою історію з 74/98-ВР документа 1998 року, якій містив 28 статей, до поточного документа 2024 року 2807-IX, який містить вже 15 статей. Головним чином НПІ визначає протоколи запуску та термінації програм які містять наступні функції: експертизи, аналізу, формування, контролю, виконання, звітування програм інформатизації на державному (галузеві програми) ті місцевому (самоврядування) рівні, а також визначає суб єктів інформатизації: генеральний замовник — Міністерство цифрової трансформації, Керівник — посадова особа, виконавці та підрядники. НПІ є власником і розробником системи обліку таких програм. НПІ визначає процеси розробки згідно ISO/IEC 12207 та ISO 9001, а супроводу та підтримки згідно ISO/IEC/IEEE 14764:2022.

2.1 Загальні принципи

На мета рівні неперервний процес реформування ОВВ зараз уявляється мені, як такий що керується наступними правилами: 1) Ін єктивність управління юстиції (як необхідний атрибут кожного міністерства, аудит, розслідування); 2) ІТ-департамент або управління (у якості зовнішнього ЄДРПО, холдер продуктів Міністерства); 3) Управління трансформації (яка механізує процес ІТ-продуктами ІТ-департаменту). Далі управління Міністерства додаються в залежності від функцій Міністерства, але ці три

плюс патронатна служба — обов'язкові. Міністерство Юстиції, Судова система, генеральна прокуратура, Поліція та інші агенції, як НАЗК, НАБУ, мають безпосередній або опосередкований доступ до (1). Це має регулюється відповідними ABAC правилами. Мінцифра має доступ до (3), як координатор мета-процесу трансформації. Також Мінцифра координує роботу і взаємодіє з (2) кожного міністерства для забезпечення каналу до реєстрів відповідних міністерств, які підтримуются відповідними IT-управліннями кожного мінстерства. Шини всіх документобігів і реєстрів координуються IT-управлінням Мінцифри, «Дія».

Завдання тансформації передбачає виконання (у тому числі) наступних цілей: 1) Кожне міністерство буде мати свій автономний і потужний IT-департамент, який обслуговує реєстри міністерства; 2) Міністерства будуть мати свої управління трансформації, працюватимуть на одному продукті, в якому будуть моделювати свою роботу; 3) Мінцифра як координатор буде затверджувати регламенти роботи IT-управлінь і управлінь трансформації кожного міністерства. «Дія» буде мати не тільки шину документообігу, продукт «Дія: Документообіг» (база), але і видавати ліцензії для учасників ринку (зараз 30 ліцензій).

2.2 Модель

1. Кабінет міністрів України
2. Міністерство освіти і науки України (МОН)
3. Міністерство охорони здоров'я України (МОЗ)
4. Міністерство внутрішніх справ України (МВС)
5. Міністерство закордонних справ України (МЗС)
5. Міністерство оборони України (МО)
6. Міністерство юстиції України (Мінюст)
7. Міністерство фінансів України (Мінфін)
8. Міністерство економіки України (Мінекономіки)
9. Міністерство енергетики України (Міненерго)
10. Міністерство соціальної політики України (Мінсоцполітики)
11. Міністерство регіональної політики України (Мінрегіон)
12. Міністерство цифрової трансформації України (Мінцифра)
13. Міністерство молоді та спорту України (Мінспорту)
14. Міністерство природних ресурсів України (Мінекології)
15. Міністерство аграрної політики України (Мінагрополітики)

2.3 Структурне ядро

1) Управління юстиції; 2) Департамент інформаційних систем виробничо-промислового управління; 3) Управління трансфор-

мації

Оскільки соціо-інформаційні системи повинні бути автономними та мати керований життєвий цикл, технічне відображення організаційної структури повинно бути під контролем міністерства, можливо у вигляді окремого підприємства (для убезпечення інтелктуальних ресурсів підприємства від ручного керування, а також оскільки IT-депратаменти міністерст відповідають за реєстри і персональні дані громадян).

Для керування структурою в реальному часі пропонується окремий вид протолу ОВВ 2.0 як наступне розширення після НПА до вже існуючого базового протоколу Документообігу згідно постанови №55 керуючого органу Кабінету Міністрів України. Наприклад, ДІТ (Державна Інституційна Трансформація).

В процесі як операційного документообігу (породжуваного структурними підрозділами міністерств), так і інституційного (породжуваного управліннями або агенціями державної цифрової трансформації) породжуються зліпки кваліфікованих електронних підписів посадових осіб які розслідуються як внутрішніми органами (відділи аудиту і відділи внутрішніх розслідувань), так і координуючим органом — Міністерством юстиції України.

2.4 Протокол державної інституційної транфсормації

Цей протокол визначає наступні операції над організаційною структурою і її політикою:

1) Створення і ліквідація структурних підрозділів; 2) Погодження та модифікація установчих конституційних документів; 3) Розробка технопроектів структурних підрозділів та їх систем; 3) Вибір і впровадження інформаційних систем для структурних підрозділів; 4) Запуск та операційна діяльність структурних підрозділів (виробництво);

Розділ 3

Органи виконавчої влади

Цей розділ описує структуру органів виконавчої влади (ОВВ), які є об'єктами інформатизації.

3.1 Міністерство науки і освіти

Ця секція є статтею-дослідженням таксономії структури Міністерства освіти і науки з точки зору як інформаційної автоматизованої системи так і соціальної структури з точки зору державного управління. Як приклад, в статті наводиться конкретна структура, яка є незначною модифікацією існуючої ієрархічної системи Міністерства освіти і науки.

3.1.1 Структурні підрозділи

1. Патронатна служба
2. Управління початкової школи
3. Управління середньої школи
4. Управління вищої школи
5. Управління юстиції
   - Департамент експертизи і сертифікації
   - Інститут інтелектуальної власності
   - Департамент кадрового забезпечення
   - Департамент аудиту і внутрішніх розслідувань
   - Департамент архівної справи (SCAN)
   - Технічний департамент (CA)
   - Департамент соціального і гуманітарного забезпечення
   - Відділ кадрів (ACC)
   - Юридичний департамент
   - Департамент міжнародного співробітництва
6. Управління науково-дослідними інститутами (агенція)
7. Національна академія наук (агенція)
   - Інститут формальної математики
     - Ректорат формальної філософії

    - Ректорат чистої математики
    - Ректорат прикладної математики
    - Ректорат мовного забезпечення
    - Ректорат теоретичної інформатики
  - Інститут формальної літератури
  - Інститут музики, кіно і образотворчого мистецтва
    - Національна консерваторія
    - Національна академія мистецтв
    - Національна кінематика
  - Інститут фізики і матеріалів
  - Інститут геології і геохімії
  - Інститут хімії і біології
  - Інститут соціальних і гуманітарних наук
    - Ректорат філософії
    - Ректорат археології
    - Ректорат національної історії
    - Ректорат права
    - Національна бібліотека
8. Управління політиками і структурними підрозділами (агенція)
  - Департамент комунікації (відділ кадрів)
  - Департамент контролю виконання показників (CRM)
  - Департамент планування переходу (аналіз процесів BPMN)
  - Департамент трансформації (широкий спектр спеціалізацій)

3.2 Міністерство охорони здоров'я

3.3 Міністерство внутрішніх справ

3.4 Міністерство закордонних справ

3.5 Міністерство оборони

Ця секція є дослідженням таксономії структури Міністерства оборони з точки зору як інформаційної автоматизованої системи так і соціальної структури з точки зору державного управління. Як приклад, в статті наводиться конкретна структура, яка є незначною модифікацією існуючої ієрархічної системи Міністерства оборони України, підсилена повним спектром інституцій для організації неперервного науково-освітнього і технологічно-виробничого процесу існування державного органу виконавчої влади — Міністерства оборони України.

У якості моделі гранулярності використана українська державна модель (міністерство, управління, департамент, відділ, сектор). Оскільки дана робота зосерджена в першу чергу на логіці існування процесу, тут значною мірою надається перевага формальним моделям, які потребують мінімальних зусиль для верифікації, моделювання і прогнозування. Оскільки формалізація процесу безпосередньо торкається інформаційного програмного забезпечення на користь приходять міжнародні стардарти телекомунікаційних протоколів, сертифікація яких торкається (в свою чергу) університетів, науково-дослідних інститутів, науково-виробничих інститутів. Науково-виробничі інститути використовується в широкому смислі як ті, що можуть бути комерційними угруповуваннями, міжнародними фундаціями, тощо. Для підтримки автономної діяльності ці всі інституції повинні входити в арсенал функціональних можливостей міністерства, включаючи головним чином університет четветого рівня акредитації, навчальні програми якого представлені частково обраними курсами п яти кафедр інституту математики НАН (див. Додатку 1). Розміщення інформаційної структури розгалуженої національної структури міністерства і його частин передбачає автономне забезпечення класу EDGE офіс з власним ресурсами охолодження, водо-електро-постачання, силами та засобами оборони.

Інформаційна політика формального моделювання передбачає довільне використання мовних сучаних засобів здатних до формальної верифікації (наявність промислових верифікаторів) TЗІ рівня Г7 (повна математична верифікація) покладаючись основним чином на телекомунікаційні протоколи і міжнародні ISO стандарти (див. Додаток 5).

3.5.1 Мотивація

Основна мотивація даної роботи полягає у висвітленні таксономії Міністерства оборони України з точки зору оптимізації, автономності існування (sustainability), само-відтворюваності, підтримки життєвого циклу існування Міністерства.

3.5.2 Модель

1. Патронатна служба
2. Управління освіти і науки (агенція)
   — Університет четвертого рівня акредитації (див. Додаток 1)
   — Науково-дослідні інститути
   — Науково-виробничі інститути
3. Управління медицини (агенція)
   — Клінічні наукові дослідження та лабораторії, НДІ ПВМ (MED)
   — Реабілітації («Пуща-Водиця», «Трускавецький», «Хмельник»)
   — Клінічні, мобільні (4) лікарні, госпіталі (14)
   — Медичні служба (5 родів), тактична медицина, медичні сили
   — Інститут медицини (ЗДМУ, ХНМУ, ЛНМУ, ТНМУ)
4. Виробничо-промислове управління (агенція)
   — КБ (ДАТ «Укроборонпром», ДАХК «Артем»)
   — Департамент економічного моделювання і планування
   — Департамент ресурсного забезпечення (SCM, TMS, WMS)
   — Департамент бюджетування і закупівель (FIN)
   — Департамент будівництва і архітектури, ліній виробництва
   — Телекомунікаційний департамент інформаційних систем
      — Відділ систем врядування
      — Відділ облікових систем
      — Відділ телекомунікаційних систем
      — Відділ безпекових протоколів Інтернет
   — Департамент авіації, авіоніки, аеронавтики і безпілотних літаючих апаратів і їх систем
      — Відділ авіації
      — Відділ авіоніки
      — Відділ аеронавтики
      — Відділ безпілотних систем
   — Департамент машинобудування (terrain) (SolidWorks)
   — Департамент кораблебудування
5. Управління юстиції
   — Відділ експертиз і сертифікації (ISO/IETF)
   — Департамент архівної справи (SCAN)
   — Департамент аудиту і внутрішніх розслідувань
   — Технічний департамент (CA)
   — Департамент соціального і гуманітарного забезпечення
   — Відділ кадрів (ACC)
   — Юридичний департамент
   — Департамент міжнародного співробітництва
6. Управління розвідки
7. Головне управління позиційною політикою
   — Мобілізаційний департамент
   — Департамент навчальних програм
   — Департамент сил і засобів оборони CRM (див. Додаток 2)

    - Родина сухопутних військ
    - Родина повітряних сил
    - Родина воєнно-морських сил
    - Родина спеціальних сил (ССО, МС, РЕБ, РХБЗ, ТРО)
    - Родина кібербезпеки і ДШВ
  - Департамент контролю і управління DFR (див. Додаток 4)
  - Економічний департамент
    - Відділ ресурсного забезпечення (WMS)
    - Відділ бюджетування і закупівель
8. Управління політиками і структурними підрозділами (агенція)
  - Департамент комунікації (відділ кадрів)
  - Департамент контролю виконання показників (CRM)
  - Департамент планування переходу (аналіз процесів BPMN)
  - Департамент трансформації (широкий спектр спеціалізацій)

3.5.3 Принципи

Одним з головних принципів закладених у фундамент МО є принцип вищої освіти, яка здобувається згідно до вимог міжнародних стандартів. Для забезпечення потреб існування працівників всіх сфер цієї таксономії в основу її неперервної маніфестації покладено існування університету четвертого рівня акредитації з усіма спеціальностями необхідними для покриття потреб самого МО.

Одним з універсальних принципів закладених в фундамент МО є принцип розподілу влади, з якого випливає три корпуси МО: 1) Політичний корпус (головне управління, перехідне управління, патронатна служба), 2) Виконавчий корпус (Управління освіти і науки, Медичне управління, Виробничо-промислове управління), 3) Судовий корпус (Управління юстиції, трибунал). Політичний корпус передбачає виділення окремої агенції яка здійснює запуск процесів створення і апробація структурних підрозділів міністерства. Сюди також входить структурний підрозділ — головне управління яке управляє силами (ЗСУ) та засобами оборони. Виконавчий корпус убезпечує інтелектуально-ємні структурні підрозділи і виокремлює їх під автономний контроль агенції з більшою дотичністю до зовнішніх структур. Судовий корпус — окремий стурктурний підрозділ з процесами аудиту і внутрішніх розслідувань в існуючих соціально-інформаційних системах в структурі МО.

3.5.4 Структурні підрозділи

1. Розподіл влади і мінімізація таксономії
2. Нормалізація формальних процесів
3. Аудит міністерства і процес трансформації

4. Архітектура структурних підрозділів
5. Апробація результатів і циклічність мета-процесу

Розподіл влади є головним принципом ефективного управління. Контролюючі органи повинні спеціалізуватися на розслідуваннях і бути виокремлені. Політики процесів і реквізитної інформації повинні здійснюватися політичним органом. Політика повинна здійснюватися окремими виконавчими органами (освіта, наука, виробництво).

Вимоги до документування процесів повинні бути на найвищому рівні (еквіваріантні семантики, спрощення процесів до нормальних форм, мінімазія горизонтальних зв язків), відповідати вимогам постанови №55 КМУ і наказу №124 МО. Повинна бути організований процес історіографії і аудиту діло-процесів згідно стандарту ISO-19510 для аудиту NATO. Запуск діло-виробництва передбачається поступово і гранулярно, спочатку від головних і малоресурних проєктів управління політик і далі згідно стратегії управління по іншим структурним підрозділам.

3.5.4.1 Патронатна служба

Сприяння реалізації політичних цілей Міністра, консультування Міністра, організаційне, інформаційне, експертно-аналітичне забезпечення діяльності Міністра.

3.5.4.2 Управління освіти і науки (агенція)

Науковий процес крім освітнього видіяє науково-дослідний і наково-виробничий у тих сферах, яких потребують департаменти організаційної структури МО. Ці компанії повинні знаходитися як і університет у сфері впливу МО.

3.5.4.3 Виробничо-промислове управління (агенція)

Пропонується повне дублювання сфер виробництва засобів оборони у відповідні департаменти, так як це є основними ресурсами головного управління тому доцільно зберігати бачення повної картини під ієрархією МО. Існуючі виробничі процеси зосереджені в «Укроборонпром» і «Артем», пропонується розглядати як зовнішні конструкторські бюро, в одному переліку з комерційними підприємствами які працюють можливо навіть проектно.

3.5.4.4 Управління юстиції

Управління юстиції бере на себе функції обліку сил (відділ кадрів) та протокольних дій в середині системи, які аналізуються антикорупційним відділом департаменту аудиту і внутрішніх розслідувань. Тут також зосереджені юридичні функції які обслуговують

всю ієрархію МО, а також зовнішніх партнерів (NATO). Також тут зосереджений центр видачі криптографічних ключів всієї ієрархії.

3.5.4.5 Головне управління

Спрощені функції головного штабу з локальним резервом сил та засобів оборони. Головна консоль (Godot) управління підпорядковує логіці ведення позиційної політики силами та засобами оборони з прогнозування подій та ціною їх усунення. Позиційна стратегія передбачає локальний облік сил та засобів оборони.

3.5.4.6 Управління політиками і структурними підрозділами (агенція)

Формальна модель адміністративного управління переходу (або врядування) від існуючої структури до будь-якої наперед заданої (як приклад наведеної). Перехідне управління займається аудитом існуючих процесів (юридичне забезпечення) та їх трансформації в урядові та облікові системи міністерства у взаємодії з Телекомунікаційним департаментом, контролем їх виконання: впровадження, апробації, калібрування, проектний менеджмент.

Процес трансформації розділений на наступні категорії (обернено до швидкоплинності): 1) трансформація (діджіталізація) існуючих процесів без модифікації логіки, такі як діловодство; 2) створення нових процесів і технологічного забезпечення для вакантних департаментів і відділів, такі як виробничо-промислове управління; 3) планування і розвиток вакантних департаментів і управлінь (довготривалий процес, розвиток навчальних програм, побудова та підтримка життєвого циклу продуктів).

Кожна фаза процесу трансормації передбачає побудову життєвого циклу проєкту, а також продукту, який покладений в його основу, сюди входить повний набір документації: Технічне Завдання, Ескізний проєкт, Технічний проєкт, Технічна документація. Управління політиками поділяє сфери компетенції з Телекомунікаційним департаментом виробничо-промислового відділу і загальні правила документування.

Важливі, головні процеси управління політиками, як основні його функції можна класифікувати так: 1) архівна справа процесного виробництва; 2) розробка процесів запуску структурних підрозділів; 3) розробка процесів документообігу; 4) розробка процесів управління і координації; 5) розробка технологічних процесів ISO-9001.

3.5.5 Результати

В статті представлена запропонована таксономія нормалізованого міністерства з урахування міжнародного досвіду організа-

ції державних структур з чітким і прозорим принципом розподілу влади і організації глобального (стратегічного) і локального (операційного) контекстів для виконання своїх функції у найпростіший і безпосередній спосіб мінімізуючи кількість протоколів взаємодії всередині системи.

В додатках статті представлена таксономія освітніх програм університету четвертого рівня акредатації і таксономія телекомунікаційного депратемнту, у функції якого входять розробка і впровадження системи. В процесі виконання завдання первинного моделювання було здійснено намагання охопити ключові органи, аж до рівня гранулярності відділів і секторів (побудова первинного індексу) і їх горизонтальних протоколів взаємодії.

3.5.6 Бібліографія

ДСТУ 2732-94 Діловодство і архівна справа.
26.05.2014 №333 Інструкція з обліку особового складу.
21.11.2017 №608 Порядок проведення службового розслідування.
26.07.2018 №370 Інструкція з діловодства.
07.04.2017 №124 Інструкція з діловодства.
07.10.2015 №393 Положення Про Юридичну Службу.
29.11.2018 №604 Інструкція з надання доповідей і донесень про події, кримінальні правопорушення, військові адміністративні правопорушення та адміністративні правопорушення, пов'язані з корупцією, порушення військової дисципліни та їх облік.

3.6 Міністерство юстиції

3.7 Міністерство фінансів

3.8 Міністерство економіки

3.9 Міністерство енергетики

3.10 Міністерство соціальної політики

3.11 Міністерство регіональної політики

3.12 Міністерство цифрової трансформації

3.13 Міністерство молоді та спорту

3.14 Міністерство природних ресурсів

3.15 Міністерство аграрної політики

Розділ 4

Специфікація та сертифікація

4.1 Законодавча база

4.1.1 Загальні положення

Базова версія «MIA: Документообіг» керується наступними загальними положеннями які виражені законами України:

— 2657-XII, Про інформацію[1],
— 7498-ВР, Про Національну програму інформатизації[2],
— 39396-ВР, Про звернення громадян[3],
— 2939-VI, Про доступ до публічної інформації[4]
— 2155-VIII, Про електронні довірчі послуги[5],
— 851-IV, Про електронні документи та електронний документообіг[6],

та розпорядженнями і постановами Кабінету Міністів України:

— 386-2013-р, Розпорядження КМУ #3860-Р [7],
— 373-2006-п, Постанова КМУ #373 [8].

[1] https://zakon.rada.gov.ua/laws/show/2657-XII
[2] https://zakon.rada.gov.ua/laws/show/74/98-вр
[3] https://zakon.rada.gov.ua/laws/show/393/96-вр
[4] https://zakon.rada.gov.ua/laws/show/2939-17
[5] https://zakon.rada.gov.ua/laws/show/2155-19
[6] https://zakon.rada.gov.ua/laws/show/851-15
[7] https://zakon.rada.gov.ua/laws/show/386-2013-р
[8] https://zakon.rada.gov.ua/laws/show/373-2006-п

4.1.2 Базова версія «MIA: Документообіг»

Продукт «MIA: Документообіг» в основному базується на Поставі #55 Кабінету Міністрів України та інших постановах КМУ:

— 55-2018-п, КМУ. Постанова #55 Деякі питання документування управлінської діяльності [9],
— 749-2018-п, КМУ. Постанова #749 Про затвердження Порядку використання електронних довірчих послуг в органах державної влади, органах місцевого самоврядування, підприємствах, установах та організаціях державної форми власності[10],
— v0144774-20, ДП «Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості". Наказ #144 Про прийняття та скасування національних стандартів ДСТУ 4163:2020 та ДСТУ 9031:2020[11],

але платформа продукту базується на наказаї Міністерства юстиції України, Міністерства цифрової трансформації, Міністерства освіти і науки, та Законами України:

— z1854-12, Міністерство юстиції України. Наказ #16005 Про затвердження Порядку роботи з електронними документами через систему електронної взаємодії органів виконавчої влади з використанням електронного цифрового підпису[12],
— z1039-20, Міністерство цифрової трансформації України. Адміністрація державної служюи спеціального зв'язку та захисту інформації України. Наказ #140614[13],
— z1306-11, Міністерство освіти і науки, молоді та спорту України. Наказ #1207 Про вимоги до форматів даних електронного документообігу в органах державної влади. Формат електронного повідомлення[14],
— z1421-14, Міністерство юстиції України. Наказ #1886/5 Про затвердження Порядку роботи з електронними документами у діловодстві та їх підготовки до передавання на архівне зберігання[15], — 851-IV, Про електронні документи та електронний документообіг[16],
— 8094-ВР, Про захист інформації в інформаційно-телекомунікаційних системах[17],

[9] https://zakon.rada.gov.ua/laws/show/55-2018-п
[10] https://zakon.rada.gov.ua/laws/show/749-2018-п
[11] https://zakon.rada.gov.ua/rada/show/v0144774-20
[12] https://zakon.rada.gov.ua/laws/show/z1854-12
[13] https://zakon.rada.gov.ua/laws/show/z1039-20
[14] https://zakon.rada.gov.ua/laws/show/z1306-11
[15] https://zakon.rada.gov.ua/laws/show/z1421-14
[16] https://zakon.rada.gov.ua/laws/show/851-15
[17] https://zakon.rada.gov.ua/laws/show/80/94-вр

4.1.3 Розширення та додаткові модулі

Розширення «MIA: Провадження»

— 4651-VI, Кримінальний процесуальний кодекс України[18],
— v0298905-20, Офіс генерального прокурора. Наказ #298 Про затвердження Положення про Єдиний реєстр досудових розслідувань, порядок його формування та ведення[19],

Розширення «MIA: Закупівлі»

— 922-19, Закон України про публічні закупівлі[20], — 169, Постанова КМУ #169,
— 1178 Постанова КМУ #1178,
— 808-20 ЗАКОН УКРАЇНИ ПРО ОБОРОННІ ЗАКУПІВЛІ,
— 1275-2022-п КАБІНЕТ МІНІСТРІВ УКРАЇНИ ПОСТАНОВА #1275,
— 1070-2019-п КАБІНЕТ МІНІСТРІВ УКРАЇНИ ПОСТАНОВА #1070,
— 224-2020-п КАБІНЕТ МІНІСТРІВ УКРАЇНИ ПОСТАНОВА #224,
— 710-2016-п КАБІНЕТ МІНІСТРІВ УКРАЇНИ ПОСТАНОВА #710,
— z0500-20 МІНІСТЕРСТВО РОЗВИТКУ ЕКОНОМІКИ, ТОРГІВЛІ ТА СІЛЬСЬКОГО ГОСПОДАРСТВА УКРАЇНИ. НАКАЗ #708,
— 544-2016-п КАБІНЕТ МІНІСТРІВ УКРАЇНИ ПОСТАНОВА #544,
— v1749731-15 МІНІСТЕРСТВО ЕКОНОМІЧНОГО РОЗВИТКУ І ТОРГІВЛІ УКРАЇНИ #1749,
— 1495-п КАБІНЕТ МІНІСТРІВ УКРАЇНИ ПОСТАНОВА #1495,

Розширення «MIA: Зброя»

— 5708, Проект Закону про право на цивільну вогнепальну зброю[21], — 5709, Проект Закону про внесення змін до Кодексу України про адміністративні правопорушення та Кримінального кодексу України для реалізації положень Закону України "Про право на цивільну вогнепальну зброю"[22],

[18] https://zakon.rada.gov.ua/laws/show/4651-17
[19] https://zakon.rada.gov.ua/laws/show/v0298905-20
[20] https://zakon.rada.gov.ua/laws/show/922-19
[21] http://w1.c1.rada.gov.ua/pls/zweb2/webproc4_2pf3516=5708skl=10
[22] http://w1.c1.rada.gov.ua/pls/zweb2/webproc4_2pf3516=5709skl=10

Розширення «MOD: Військова частина»

— ДСТУ 2732-94, Діловодство і архівна справа.
— 26.05.2014 #333, Інструкція з ведення обліку особового складу.
— #608 Порядок проведення службового розслідування.
— 26.07.2018 #370, Інструкція з діловодства.
— 07.04.2017 #124, Інструкція з діловодства.
— 07.10.2015 #393, Положення Про Юридичну Службу.
— 29.11.2018 #604, Інструкція з надання доповідей і донесень про події, кримінальні правопорушення, військові адміністративні правопорушення та адміністративні правопорушення, пов'язані з корупцією, порушення військової дисципліни та їх облік.

4.2 Класифікація вимог

4.2.1 Вимоги до інтерфейсу користувача

4.2.2 Вимоги до адміністрування системи

4.2.3 Вимоги типових ділопроцесів системи

4.2.4 Вимоги процесінгової системи

4.2.5 Вимоги інтеграції з зовнішніми системами

4.2.6 Вимоги до розподіленої роботи

4.2.7 Вимоги до комплексу засобів захисту (КЗЗ)

4.2.8 Технічні вимоги до зберігання даних

4.3 Відповідність міжнародним стандартам

4.3.1 Стандарти RFC

4.3.2 Стандарти ISO

4.3.3 Національні стандарти ДСТУ та NIST

4.4 Засоби захисту та ступені гарантії безпеки

4.4.1 Мануальна наочна верифікація

4.4.2 Інтеграційне тестування

4.4.3 Математична верифікація

Розділ 5

Державна система

По аналогії зі стандартом ISO 42010 «Фреймворку Закмана», фреймворк Максима Сохацького визначає та уточнює архітектурні рівні з яких складаються сучасні корпоративні інформаційні системи:

— Юридично-документальний рівень
— Обліково-реєстровий рівень
— Зв'язність людей та пристроїв
— Телекомунікаційна платформа
— Схема та метаінформація
— Безпека інтернету

5.1 Юридично-документальний рівень

Згідно фреймворку верхній шостий рівень визначає BPMN процеси згідно яких здійнюється відзеркалення юридично-правивих відносин електронного документообігу. Кожен крок такого процесу, та усі його документи підписуються особистим ключем КЕП посадової особи, що дає змогу проведення диспутів та розслідувань Міністерством юстиції України. Окрім того цей рівень системи орієнтований на аналітику у взаємодії з громадянами через СЕВ ОВВ.

У 2022 році юридично-документальні системи ERP/1 будуються на сховищі з єдиним простором ключів Facebook RocksDB, що здатне працювати через Intel SPDK на NVMe дисках, наприклад у складі таких сховищ як CEPH. Обсяг обігу документів на великих підприємствах сягає 1ТБ на рік.

5.2 Обліково-реєстровий рівень

Обліково-реєстровий рівень пропонує низькорівневе масштабоване розподілене журнальне сховище даних та метаданих, яке може бути побудоване на реляційних базах даних, базах даних з єдиним простором ключів з гарантіями консистентності (chain-hash) або їх комбінаціях.

Класичні представники цього рівня в системах управління підприємствами: система управління людськими та матеріальними ресурсами, банківські системи PCI DSS, складські системи, системи управління поставками та виробництвом, системи сервісних послуг, системи управління проектами, тощо.

5.3 Зв'язність людей та пристроїв

Рівень зв'язності людей та пристроїв визначає комунікаційні протоколи та технології, які об'єднують головні ресурси підприємства (пристрої та людей) у одну телекомунікаційну мережу. Як правило виробництво складається з багатьох пристроїв що підключаються до промислових шин як MQTT, та робочих місць користувачів.

З точки зору продуктів цей рівень представляється зазвичай корпоративними комунікаторами та дашбордами де здійнюється моніторинг роботизованого обладнання: пристрої, датчики, тощо. Ресурси підприємства — люди та пристрої як правило зберігаються в LDAP директорії підприємства.

5.4 Телекомунікаційна платформа

Рівень платформи визначає засоби мастшабування пам'яті (персистентної та волатильної) та обчислювальних ресурсів (за допомогою процесінгових брокерів доставки повідомлень). Це рівень визначає реляційні бази даних та бази даних з єдиним простором ключів, а також стандарти та протоколи передачі інформації у промислових ERP системах, такі як CSV, JSON, SOAP, BERT, ASN.1, тощо.

5.5 Схема та метаінформація

Рівень схеми даних визначає модель зберігання даних як з точки зору об'єктів-сутностей та і з точки зору технологій та протоколів, які необхідні для їх опису. Головним чином це Фреймворк Закмана та сімейство стандартів які описують UML.

5.6 Безпека інтернету та інфраструктури

Рівень безпеки визначає схему функціонування основного центрального засвідчувального орнагу, акредитованих центрів сертифікації ключів, протоколи шифрування та підпису, директорію підприємства, інтернет протоколи найменування ресурсів. Усе визначено згідно ASN.1 специфікації. Компанія ІНФОТЕХ є утримувачем та автором усіх імплементації.

Розділ 6

Юридично-документальний рівень

6.1 Вступ

Друге видання КНИГИ ERP (англ. ERP BOOK VOL.3 Blue Book) визначає формальну бізнес специфікацію та її імплементацію для сучасних оптимізованих підприємств. Системи ERP на її базі також уже не один рік використовується у банківській сфері, процесінгу транзакцій, розподілених системах повідомлень, в IoT секторі.

6.2 Модулі підприємства

ERP/1 є комплексом бібліотек (N2O.DEV) та підсистем додатків (ERP/1), який використовує загальну шину і загальну розподілену базу даних для швидкіснх операційних вітрин.

ERP — Даний модуль обліково-реєстраційного рівня зберігає основну ієрархічну структуру підприємства, її схему, метаінформацію про типи даних, а також сам інформіцію: записи про персонал, інвентар, компанії та офіси підприємства.

CRM — Система управління зв'язками з громадськістю та органами виконавчої влади: являє собою базову реалізацію постанови #55 КМУ.

CART — Система управління клієнтами: являє собою розширення більш абстрактного додатку CHAT.

6.3 Управління ресурсами

Головним чином інформаційна структура нашого підприємства складається з обчислювальних ресурсов (додатки, запущені в шині) та накопичувальних ресурсів (дані, збережені в базі даних). SOA архітектура в якості моделі управління обчислювальними ресурсами пропонує асинхронний протокол віддаленого виклику на шинах. Разом з N2O можно використовувати MQTT та інші шини, за допомогою наступних протоколів: TCP, WebSocket. Ці асинхронні протоколи часто називають протоколами реального часу, оскільки в них функції відправки повідомлень завжди миттєво повертають результат. Що ж стосується протоколів для публікації і доступу до даних, то тут може виявитися доречним використання синхронного HTTP протоколу.

6.4 Архітектура CRM системи

6.4.1 Сторінки

Перелік сторінок

```
def route(<<"ldap", _::binary>>), do: LDAP.Index
def route(<<"crm",  _::binary>>), do: CRM.Index
def route(<<"rmk",  _::binary>>), do: RMK.Index
def route(<<"kvs",  _::binary>>), do: KVS.Index
def route(<<"act",  _::binary>>), do: BPE.Actor
def route(<<"help", _::binary>>), do: HELP.Index
```

6.4.1.1 LDAP

Сторінка авторизації користувачів.

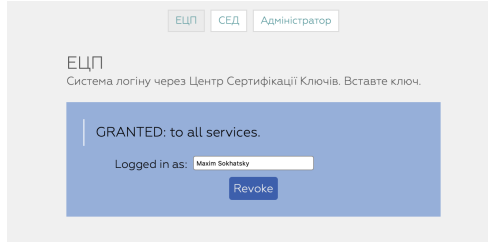


Рис. 6.1 Сторінка авторизації

6.4.2 Комболукап

6.4.3 Сервіси

6.4.4 CEB OBB

6.4.5 Шаблони

6.4.6 Дерева

6.4.7 Процеси

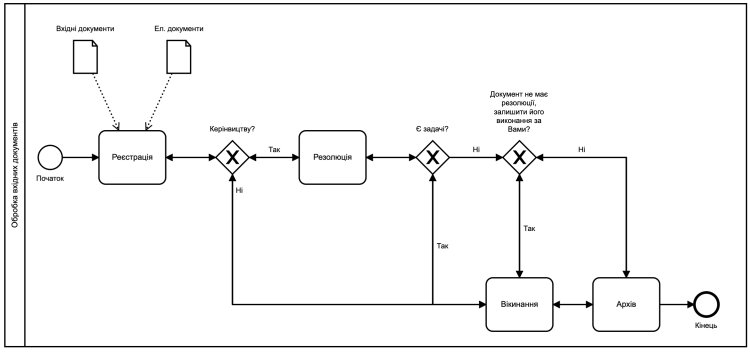
Даний модуль інкапсулює визначення Схеми, Бізнес-Процесів та Форм, які використовуються у системі Infotech-ERP згідно методології фреймворку Захмана.

6.4.7.1 Формування нормативно-довідкової інформації

Виділяються наступні основні процеси організаційно-розпорядчих документів: «Накази», «Протоколи», «Доручення керівництва».

6.4.7.2 Обробка вхідних документів

Вхідні документи надходять в УДСД, ВОРЗГ та ВОДПІ. При надходженні документа уповноважена посадова особа зазначених СП реєструє його в системі та виконується його подальша обробка.



Бізнес-процес обробки вхідних докумнетів Рис.

Далі вхідний документ надходить або Міністру/Заст. Міністра/-Державному секретарю для накладання резолюції, або до СП на виконання.

6.4.7.3 Вхідні документи

6.4.7.4 Вихідні документи

Вихідні документи створюються в підрозділах (ініціатор документа). Вихідні документи можуть виникати з ініціативи співробітників Міністерства або ж в результаті обробки вхідних документів. Якщо вихідний документ пов'язаний з вхідним документом, то відповідальному працівнику необхідно вказати посилання на пов'язаний документ. Обробка документів виконується по одному бізнес-процесу, незалежно від місця виникнення документа.

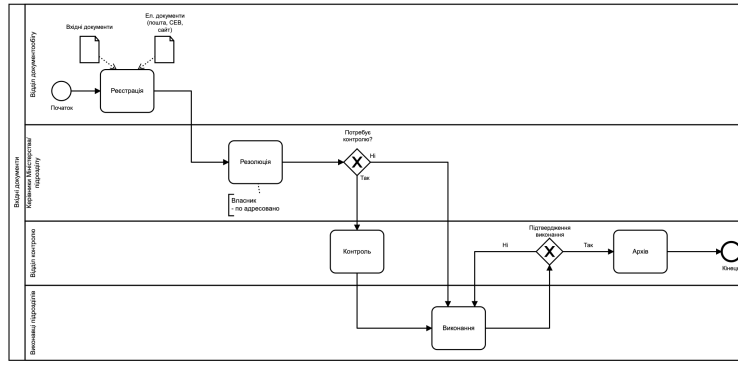


Рис. 6.3 Бізнес-процес вхідних докумнетів

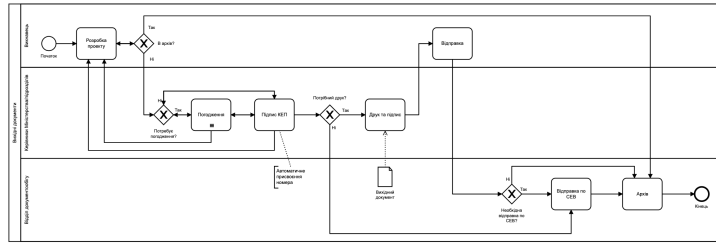


Рис. 6.4 Бізнес-процес вихідних докумнетів

В системі реєструється проект вихідного документа, який повинен бути погоджений з переліком погоджуючих осіб. Після підпису фінальним підписантом, документу присвоюється номер та виконується відправка.

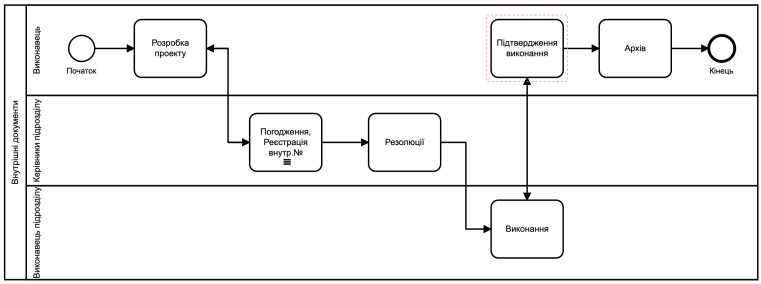
6.4.7.5 Внутрішні документи

Внутрішні документи можуть вводитися всіма учасниками документообігу. Виділяються наступні основні бізнес-процеси внутрішніх документів: «Доповідна записка», «Лист».

6.4.7.6 Організаційно-розпорядні документи

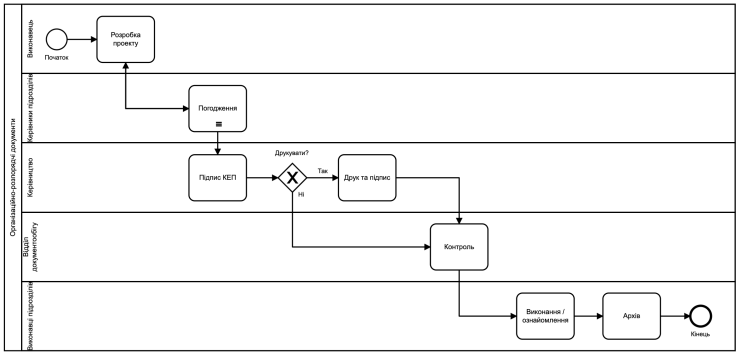
Розробка проекту документа

На даному етапі розробляється електронний проект документа: заповнюються всі необхідні реквізити в електронній картці документа, після збереження електронної картки автоматично прикріплюється шаблон документа як оригінал. Виконавець вносить вміст документа в оригінал і зберігає його. Ініціатор додає всіх виконавців, кому адресований наказ. По полю «Адресовано» ав-



Бізнес-процес внутрішніх докумнетів

Ри



Бізнес-процес організаційно-розпорядних докумнетів

Рис. 6.6

томатично будуть створені задачі на виконавців. Далі необхідно передати документ на наступний крок.

Погодження

На даному кроці виконується погодження документа особами, які були вказані Виконавцем при створенні проекту документа. Обов'язковою умовою передачі документа на наступний етап - позитивне погодження від ВСІХ погожуючих осіб. Інакше далі передати документ неможливо. Якщо один з візуючих відхилив документ (при цьому вноситься коментар з причинами відхилення і зауваженнями до документа) - в даному випадку документ повертається на першу стадію Виконавцю на доопрацювання. Якщо всі особи погодили документ - він автоматично передається на наступну стадію. Після погодження документу можна сформувати Аркуш погодження у вигляді друкованої форми.

КЕП

На даній стадії документ підписується в електронному вигляді Керівництвом Міністерства. Під час цього документу присвоюється реєстраційний номер. У разі налагодження СЕД на використання QR-коду, він розміром 21 на 21 мм розміщується в нижньому лівому куті першої сторінки документа. У разі налагодження СЕД на використання штрих-коду, він розміщується у правому кутку нижнього поля першої сторінки документа.

Підпис

Після підписання організаційно-розпорядчого документу, у разі необхідності створення паперового варіанту уповноважена особа служби (помічник) Міністра/заступника міністра/державного секретаря роздруковує документ та надає на підпис керівнику. Якщо організаційно-розпорядчийдокумент підписано керівником підрозділу, то при необхідності документ роздруковує виконавець.

Постановка на контроль

На даному етапі контролюючий СП перевіряє завдання по документу, при необхідності здійснює постановку на контроль, та періодичність. Документи і задачі на контролі незалежно від кроку опрацювання документа доступні за окремим фільтром, їх можна відстежувати незалежно від того, на якій стадії знаходиться документ, контролювати виконання, проводити аналіз, і т.д.

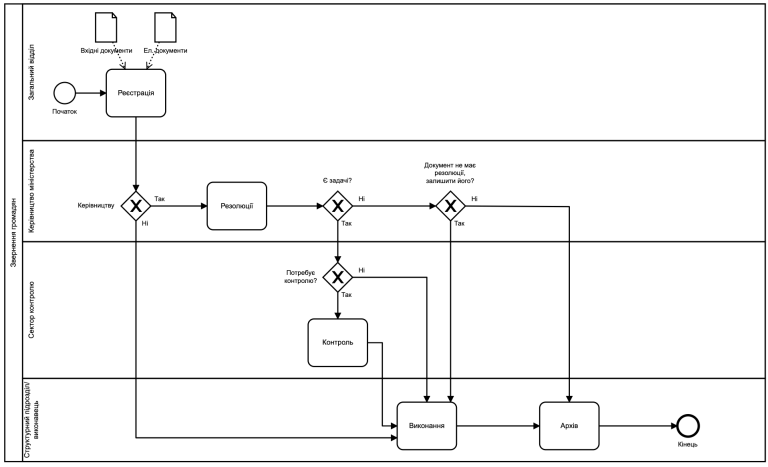
Виконання/Ознайомлення

На даному етапі контролюючий СП перевіряє завдання по документу, при необхідності здійснює постановку на контроль, та періодичність. Документи і задачі на контролі незалежно від кроку опрацювання документа доступні за окремим фільтром, їх можна відстежувати незалежно від того, на якій стадії знаходиться документ, контролювати виконання, проводити аналіз, і т.д.

Цифровий шифровий архів

Після ознайомлення документ переходить в Архів, який додатково накладає підписи КЕП Архіву та утримує істрорію всіх проміжних сертифікатів АЦСК до ЦЗО..

6.4.7.7 Звернення громадян



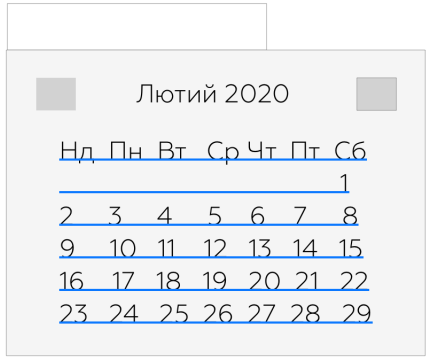
Бізнес-процес звернення громадян

6.4.8 Елементи

Тут зібрана мінімальна кількість бізнес-форм, специфічних для CRM СЕД, яка необхідня для забезпечення реалізації функціональних вимог замовника.

6.4.8.1 Календар

Календар взятий з бібліотеки NITRO, проте потребує додаткової стилізації.



Контрольний елемент Календар

6.4.8.2 Пошук по довільним фідам

Для забезпечення пошуку по словникам та бізнес-об'єктами системе передбачається створення спеціалізованого скалярного комбо-пошуку по довільним фідам в сховищі даних. Наприклад: Співробітники, Населені пункти КОАТУУ, тощо.

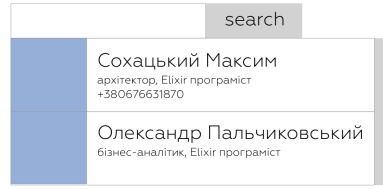


Рис. 6.9 Контрольний елмент віддаленого пошуку по базі даних

6.4.8.3 Форма редагування та пошуку

Для кожного типу документу в системі реєструються дві форми: форма пошуку та форма редагування (вона ж форма створення нового). Наявність двох форм вмотивована відмінністю валідаторі: для пошуку валідатори повинні дозволяти пусті поля, позаяк для редагування валідатори повинні перевіряти валідність полів бізнес-об'єктів.

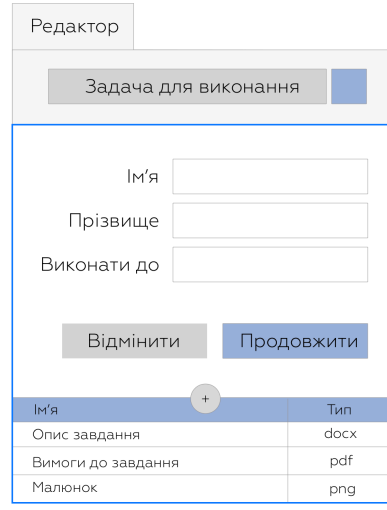


Рис. 6.10 Контрольний елемент редагування документу та підлеглих файлів

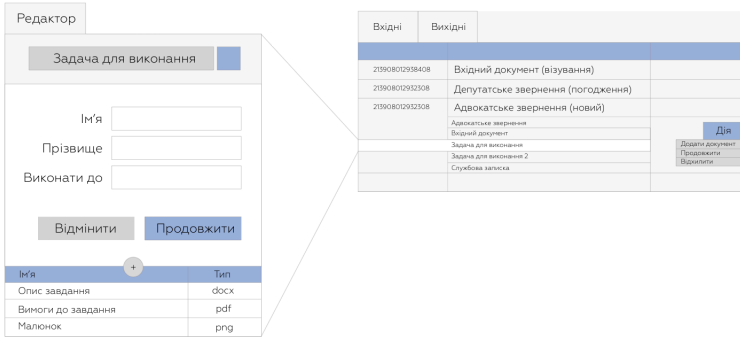
6.4.8.4 Управління бізнес процесом

Для управління завдання, доступу до документів процесу, створення нових документів в процесі, візування, підпису, проштовху документів по бізнес-процесу використовується стандартний контрольний елемент управління бізнес-процесом.

Контрольний елемент управління бізнес-процесами Рис. 6.11

6.4.8.5 Документи в бізнес-процесах

При навігації по дукументам процесу передбачається миттєве відображення підлеглого документа в лівій панелі головної сторінки користувацького інтерфейсу.



Навігація по документам бізнес-процесу Рис.

6.4.8.6 Використання контролів на формах

Приклад використання контрольного елементу довільного пошуку на формах.

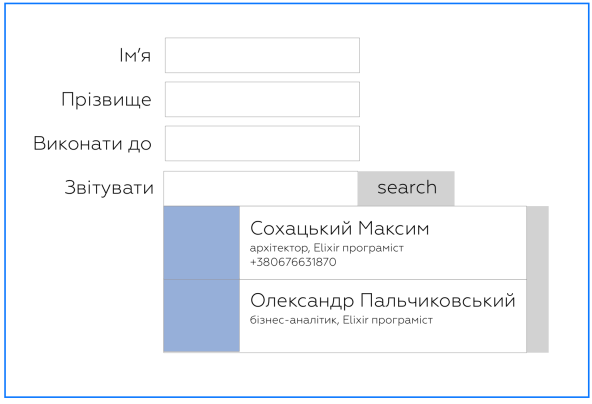


Рис. 6.13 Приклад використання контрольних елементів на формах

6.4.8.7 Контрольний елемент КОАТУУ

Приклад використання контрольного елементу КОАТУУ.

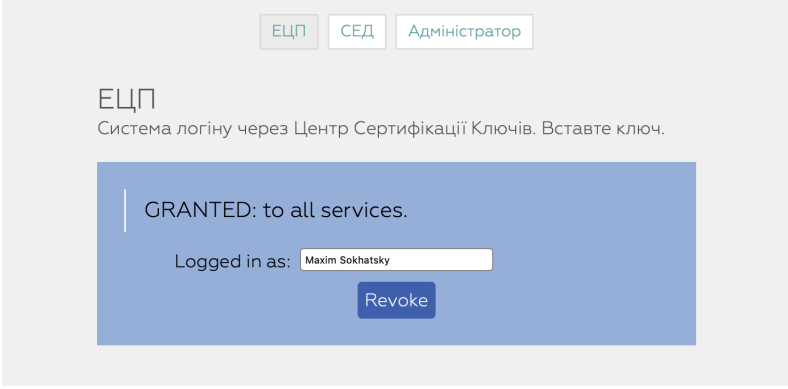
Рис. 6.14 Приклад використання контрольного елементу КОАТУУ

6.4.9 Редактори

Тут будуть перелічені контроллери сторінок, кожна з яких є SPA веб додатком.

6.4.9.1 Вхід в систему

Сторінка входу в систему з використанням ЕЦП.



Сторінка входу в систему

6.4.9.2 Робота з документами

Головна сторінка системи для роботи з документами в бізнес-процесах.

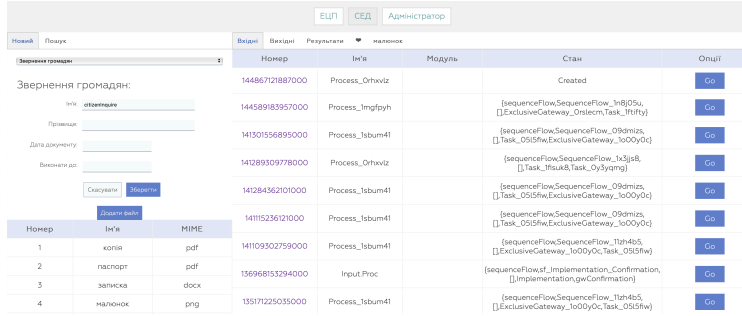


Рис. 6.16 Сторінка роботи з документами

При навігації по дукументам процесу передбачається миттєве відображення підлеглого документа в лівій панелі головної сторінки користувацького інтерфейсу.

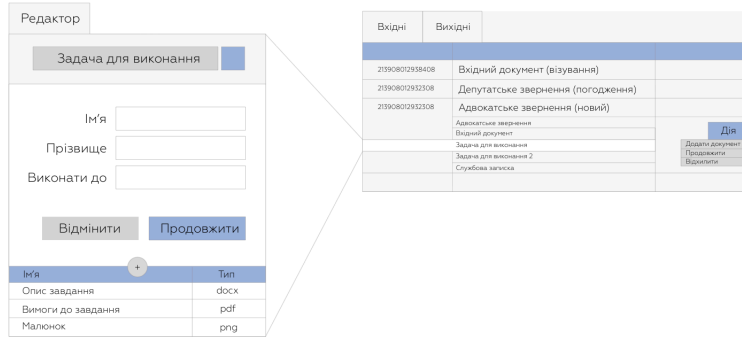


Рис. 6.17 Навігація по підлеглим документам

6.4.10 Конструктор

Тут представлені адміністративні сторінки управління системою.

6.4.10.1 Бізнес-об'єкти

Глобальний каталог усіх бізнес-об'єктів системи.

6.4.10.2 Бізнес-процеси

Перелік усіх зареєстрованих бізнес-процесів в системі, та можливість їх тестування.

6.4.10.3 Бізнес-форми

Перелік усіх форм документів та бізнес-форм користувача, зареєстрованих в системі.

Розділ 7

Обліково-реєстраційний рівень

7.1 Вступ

Друге видання КНИГИ ERP (англ. ERP BOOK VOL.3 Blue Book) визначає формальну бізнес специфікацію та її імплементацію для сучасних оптимізованих підприємств. Системи ERP на її базі також уже не один рік використовується у банківській сфері, процесінгу транзакцій, розподілених системах повідомлень, в IoT секторі.

7.1.1 Види реєстрів

1) Реєстри орієнтовані на суб'єктів організаційних систем;
2) Реєстри орієнтовані на облік матеріальних ресурсів;
3) Реєстри орієнтовані на георафічні об'єкти;
4) Реєстри орієнтовані на події;
5) Реєстри орієнтовані на документи, накази, НПА;
5) Реєстри медичних систем (FHIR);
5) Реєстри предметно-орієнтованих словників для функціональних підсистем.

7.1.2 Функціональні можливості

7.2 Модулі підприємства

ERP/1 є комплексом бібліотек (N2O.DEV) та підсистем додатків (ERP/1), який використовує загальну шину і загальну розподілену базу даних для швидкіснх операційних вітрин.

FIN — Фінансовий модуль підприємства для бухгалтерії, зберігає бізнес процеси, які представляють собою рахунки учасників системи: персонал (для нарахування зарплат), рахунки та субрахунки підприємства (для здійснення економічної діяльності) і зовнішні рахунки в платіжних системах.

ACC — Система управління персоналом: зарплатні відомості, календар підприємства, відпустки, декретні відпустки, інші календарі.

SCM — Система управління ланцюжком поставок: головний БП системи — експедиційний процес доставки товарів ланцюжку одержувачів за допомогою транспортних компаній.

PLM — Система управління життєвим циклом проектів і продуктів. Також містить CashFlow та P&L звіти.

PM — Система управління проектами підприємства з деталізацією часу і протоколів прийому-передачі (прийняті коміти в гитхабі).

WMS — Система управління складом та деталями.

TMS — Система управління транспортом підприємства.

7.3 Архітектура CART системи

Розділ 8

Технологічний рівень зв'язності людей та пристроїв

8.1 Вступ

Ця глава визначає форамальну специфікацію на програмне забезпечення усіх рівнів моделі Закмана для підприємств ISO-42010, містить широкий спектр прикладів, розказує про складові компоненти та є вичерпним авторським стартовим посібником для курсу навчання розробки технологічних програм для платформи Erlang і публікації в системі електронної взаємодії державних електронних інформаційних ресурсів "Трембіта".

Трембіта — це система побудована на пакетах Ubuntu 18 LTS і пропонує головним чином інфраструктуру X.509, а також розгортання національного форку шини X-ROAD, який використовується як сереовище для гетерогенних сервісів, в якому всі 15 міністерств публікують свої сервіси і їх клієнтські адаптери. Каталог сервісів доступний публічно[1]. Сам інтерфейс управління X-ROAD написаний з використаннями SOAP/WSDL специфікацій. В цьому цифровому просторі відбувається взаємодія (передача конфіденційних даних в основному) між обліково-реєстраційними системами міністерств на основі безпосередніх захищеник каналів зв'язку між сервісами та їх споживачами, яка містить просту і фіксовану логіку.

СЕВ ОВВ, на відміну від системи Трембіта — це публічна шина даних для урядової кореспонденції і нормативно-правових актів, вона побудована згідно ISO/IEC 11756:2010 (MUMPS) на базі продукту InterSystems Caché. В цьому цифровому просторі відбувається робота систем врядування юридично-документального рівня

[1] https://catalog.trembita.gov.ua

8.2 Виробничий процес

8.3 Системи сховищ даних

8.3.1 Реляційні бази даних

8.3.2 Бази даних з єдиним простором ключів

8.3.3 Шини комунікації та брокери повідомлень

8.3.4 Розміщені в пам'яті гарячі дані

8.4 Обчислювальні ресурси

Концептуальна модель системи в рамках якої функціонує N2O визначаена як обчислювальне середовище, яке складається з процесору подій (N2O), операційного (ETS) та персистентного сховища (KVS). З точки зору обчислювального середовища, ресурси підприємства складаються з глобального сховища та обчислень, які розділяють глобальну адресацію та представляють собою Erlang-процеси (N2O протоколи). Кожен процес PI, може містити певний набір протоколів, будь-який з яких відповідає на певний набір повідомлень. Протоколи N2O визначені на точці підключення повинні не перетинатися, в іншому випадку протокольні модулі можуть перехоплювати та впливати на інші протокольні модулі, які повинні реагувати на той самий тип повідомлень.

Усі асинхронні процеси PI запускаються під головним супервізором n2o та індексуються URI ключем разом з типом реактивного каналу реального часу: ws або mqtt. N2O протоколи підключені безпосередньо до веб-сокет точок підключення виконуються в контексті TCP процесів, у даному випадку TCP-сервера бібліотеки RANCH, супервізор ranch_sup.

```
> :supervisor.which_children :n2o
[
  { {:ws, '/chat/ws/4'}, <0.985.0>, :worker, [:n2o_ws] },
  { {:ws, '/chat/ws/3'}, <0.984.0>, :worker, [:n2o_ws] },
  { {:ws, '/chat/ws/2'}, <0.983.0>, :worker, [:n2o_ws] },
  { {:ws, '/chat/ws/1'}, <0.982.0>, :worker, [:n2o_ws] },
  { {:mqtt, '/bpe/mqtt/4'}, <0.977.0>, :worker, [:n2o_mqtt] },
  { {:mqtt, '/bpe/mqtt/3'}, <0.976.0>, :worker, [:n2o_mqtt] },
  { {:mqtt, '/bpe/mqtt/2'}, <0.975.0>, :worker, [:n2o_mqtt] },
  { {:mqtt, '/bpe/mqtt/1'}, <0.974.0>, :worker, [:n2o_mqtt] },
  { {:caching, 'timer'}, <0.969.0>, :worker, [:n2o] }
]
```

8.4.1 Накопичувальні ресурси

Розподілені хеш-кільця використовуються не тільки для розподілених обчислень, але і для зберігання даних. Деякі бази даних, наприклад RocksDB та Cassandra, використовують глобальний простір ключів для даних (на відміну від таблично-орієнтованих баз). Саме для таких баз і створено библиотеку KVS, де в якості синхронного транзакційного інтерфейсу — API ланцюжків з гарантією консистентності. Нижче наведено приклад структури ланцюжків екземпляру системи PLM:

```
> :kvs.all :writer
[
  {:writer, '/bpe/proc', 2},
  {:writer, '/erp/group', 1},
  {:writer, '/erp/partners', 7},
  {:writer, '/acc/synrc/Kyiv', 3},
  {:writer, '/chat/5HT', 1},
  {:writer, '/bpe/hist/1562187187807717000', 8},
  {:writer, '/bpe/hist/1562192587632329000', 1}
]
```

В нашій моделі синхронні протоколи використовуються для управління накопичувальними ресурсами підприємства і транзакційного процесингу.

8.5 Типові специфікації

Протоколи визначаються типовими специфікаціями і генеруються для наступних мов: Java, Swift, JavaScript, Google Protobuf V3, ASN.1. Також ми генеруємо валідатори даних по цих типових анотаціях і вбудовуємо ці валідатори в тракт наших розподілених протоколів, тому ми ніколи не дозволимо клієнтам зіпсувати сторадж. Для веб додатків у нас развинута система валідації — як для JavaScript, так і на стороні сервера. Бізнес логіка повністью ізольована в нашій системі управління бізнес процесами, де кожен бізнес процесс є процесом віртуальної машини. Всі ланцюжки модифікуються атомарним чином, підтримують flake адресацію, і не вимагають додаткової ізоляції у своєму примітивному використанні. Тому ви можете трактувати базу як розподілений кеш і використовувати її з фронт додатків для примітивних випадків.

8.6 Середовище

Для забезпечення повного замкненого середовища пропонують наступні заміни бібліотек kernel та stdlib:

🚀 VM — віртуальна машина середовища виконання[2]
☆ BASE — базова системна бібліотека як заміна stdlib[3]
☆ RT — бібліотека середовища виконання як заміна kernel[4]
📨 SYN — бібліотека PubSub для розподілених систем[5]
⚡ MAD — бібліотека управління пакетами та інстансами[6]

[2] vm.n2o.dev
[3] base.n2o.dev
[4] rt.n2o.dev
[5] syn.n2o.dev
[6] mad.n2o.dev

8.6.1 Бібліотеки

Для запезпечення повноцінної промислової специфікації ERP/1, ми розширили набір інструментальних засобів наступними бібліотеками: формальними представленнями презентаційного рівня FORM та системою управління бізнес-процесів BPE. FORM представляє собою декларативну бібліотеку побудови графічних інтерфейсів, а бібліотека BPE підтримує XML файли стандарту BPMN 2.0 та реалізує безпосередню інтерналізацію BPMN семантики у семантику віртуальної машини Erlang.

- N2O — сервер протоколів для стандартів MQTT/WS/QUIC[7]
- NITRO — UI веб-фреймворк Nitrogen[8]
- KVS — бібліотека доступу до KV сховищ RocksDB[9]
- FORM — бібліотека декларативного конструювання іформ[10]
- BPE — сисема управління процесами стандарту BPMN 2.0[11]
- RPC — бібліотека генерації SDK для мов JS, protobuf, Swift[12]

8.6.2 Приклади

Головні приклади фундації N2O.DEV присвячені наступним темам: MQTT та WebSocket чати для демонстрації веб-фреймворку NITRO, який працює як модуль N2O, приклад REST адаптер до бази даних KVS, та повністю чистий N2O додаток CHAT на основі бібліотеки SYN без використання NITRO:

- SAMPLE — ідіоматичний приклад Nitrogen поверх WS[13]
- REVIEW — ідіоматичний приклад Nitrogen поверх MQTT[14]
- REST — бібліотека для побудови HTTP API[15]
- CHAT — приклад системи доставки повідомлень[16]

[7] ws.n2o.dev
[8] nitro.n2o.dev
[9] kvs.n2o.dev
[10] form.n2o.dev
[11] bpe.n2o.dev
[12] rpc.n2o.dev
[13] sample.n2o.dev
[14] review.n2o.dev
[15] rest.n2o.dev
[16] chat.n2o.dev

8.7 Бінарні протоколи та мови їх опису

8.7.1 Мова опису протоколів ASN.1

8.7.2 Мова опису протоколів Protobuf/GRPC

8.7.3 Мова опису протоколів SOAP/XSD/XML

8.7.4 Мова опису протоколів N2O.DEV RPC

8.8 Формати передачі даних

8.8.1 Формати передачі даних ETF/BERT

8.8.2 Текстовий формат з метаописом JSON/JTD

8.8.3 Колоночний формат з метаописом CSV/CSM

8.9 Розробка Інтернет додатків

8.9.1 Erlang та сучасний веб

Erlang реалізує недосяжну мрію кожного обчислювального середовища для паралельної та узгодженої конкуретної обробки повідоблень. Так найбільш відомі бібліотеки акторів (Akka, Orleans), які реалізують основні примітиви: процесори та черги, копіюють модель акторів Erlang, зазвичай намагаються також реалізують додатково механізми перезапуску та супервізії процесів подібно до Erlang, проте тільки Erlang забезпечує soft real-time характеристики, завдяки керуванню латенсі з точністю до таймінгу команд віртуальної машини. А з виходом 24 версії в 2020 році, яка почала підтримувати JIT-компіляцію завдяки asmjit, продуктивність та чуттєвість віртуальної машини зросла ще більше.

З формальної точки зору достатньо добре ізольоване середовище віртуальної машини Erlang не тільки забезпечує характерстики реального часу для SMP-планувальника легких зелених процесів, але і обмежує область видимості heap пам'яті виключно для процесів-власників, що унеможливлює вплив відмови певних процесів на глобальний стан віртуальної машини.

Erlang ідеально підходить для побудови високо-навантажених, просто-масштабованих, подійно-орієнтованих, неблокуючих, надійних, постійно-доступних, високо-ефективних, швидких, безпечних та надійних систем обробки повідомлень та розподілених у просторі та часі систем.

8.9.2 DSL vs Шаблони

З технічної точки зору N2O успішно показує неперевершену досі якість DSL програмування, яку ви не зможете знайти в сучасних веб-фреймворках для мов Erlang та Elixir. За 7 років неперервної еволюції N2O ми переписали кожен з 700 рядків по 30 разів, якшо порахувати через коміти Github. Веб-фреймворк NITRO, сховище KVS, та BERT.JS кодування може забезпечити відображення в веб-браузері повноекранних вертикальних форм з усіма обчислюваними полями зі швидкістю 60 форм в секунду по веб-сокет каналу. А надзвичайно компакта JavaScript бібліотека-компаньйон вміщується в 4 MSS/MTU вікна — саме такий розмір мінімального веб-клієнта з BERT кодуванням, який повністю управляється зі сторони сервера.

N2O сервер та веб-фреймворк NITRO реалізують концепцію не тільки управління сесіями та каналами, але і усім стеком побудови додатків включаючи UI частину, як це відбувається у таких веб-фреймворках як Erlang Nitrogen, OCaml Ocsigen, Scala Lift,

F# WebSharper, а завдяки таким розширенням як FORM та BPE ідеально підходять і для побудови автоматизованих CRM систем.

Це не означає, що за допомогою N2O ви не можете створювати більш класичні та архаїчні додатки у стилі DTL шаблонізаторів, або як це відбувається у таких фреймворках як PHP, ASP, JSP, Rails, тощо. Перші версії NITRO містили в прикладах використання Django Template Library (DTL), проте задля чистоти стеку були прийнято не включати в N2O додаткові шаблонізатори крім NITRO DSL.

8.9.3 Історія

N2O сервер, а також NITRO веб-фреймворк були спроектовані як інструментільні засоби для створення промислових ERP модулів підприємства у складі відкритої платформи ERP/1. Напочатку, N2O був відгалужений, як оптимізована версія веб-фреймворку Nitrogen, створеного Расті Клопгаузом. Хотілося оптимізувати та вдосконалити мінімізований WebSocket-тракт, який не містить синхронного протоколу HTTP взагалі та дозволяє створювати повноцінні асинхронні веб-додатки реального часу. На ньому була створена система управління депозитами в національному банку ПриватБанк. Пізніше N2O був розділений на бібліотеку-фреймворк процесів та протоколів (власне N2O) та бібліотеку-веб-фреймворк NITRO. Бібліотеки N2O та NITRO також отримали можливість роботи не тільки через WebSocket але і через MQTT та через чисті TCP або UDP. Така оновлена версія 5.10 була впроваджена як ядро системи повідомлень для додатку NYNJA з відкритим open-source протоколом і саме ій присвячений друга версія підручника.

8.9.4 Інтерфейс NITRO

8.9.5 Сховище KVS

8.9.6 Логіка BPMN

8.9.7 Додатки MQTT та WebSocket

Розділ 9

Схема даних, типи, валідація та генерація

9.1 Графічні мови представлення метаінформації UML

9.2 Алгебраїчні мови та System F

9.3 Моделі процесів

9.4 Верифікація типів

9.5 Генерація SDK та конекторів

9.6 Базова схема підприємства ERP/1

Розділ 10

Інфраструктурний рівень безпеки інтернету

10.1 Електронний підпис і цифрова печатка

Кваліфікований Електронний Підпис, або Кваліфіклована Електронна Печатка — це набір стандартів криптографічного захисту ДСТУ 4145, та міжнародних стандартів які визначають його конверт: X.501, X.509, X.511, X.520.

Серія міжнародних стандартів X.500, групується по категоріям, кожна з яких має свій перелік ASN.1 файлів. Аби підключити усі визначення необхідні для КЕП використані наступі компоненти стандартів (виділені болдом): X.501 — BasicAccessControl [1], InformationFramework [2], UsefulDefinitions [3]; X.509 — SpkmGssTokens [4], PkiPmiExternalDataTypes [5], AttributeCertificateDefinitions [6], AlgorithmObjectIdentifiers [7],
AuthenticationFramework [8], CertificateExtensions [9];
X.511 — SpkmGssTokens [10], DirectoryAbstractService [11]; X.520 — PasswordPolicy [12], UpperBounds [13], SelectedAttributeTypes [14].

Можно було би винести необхідні визначення одразу в KEP.asn1, однак цим хотілося підкреслити сумісність з міжнародними стандартами. Окрім серії протоколів X.500, КЕП ще визначає також запити та відповіді OCSP, також у ASN.1 форматі.

[1] https://www.itu.int/ITU-T/formal-language/itu-t/x/x501/2019/BasicAccessControl.html
[2] https://www.itu.int/ITU-T/formal-language/itu-t/x/x501/2019/InformationFramework.html
[3] https://www.itu.int/ITU-T/formal-language/itu-t/x/x501/2019/UsefulDefinitions.html
[4] https://www.itu.int/ITU-T/formal-language/itu-t/x/x509/2019/ExtensionAttributes.html
[5] https://www.itu.int/ITU-T/formal-language/itu-t/x/x509/2019/PkiPmiExternalDataTypes.html
[6] https://www.itu.int/ITU-T/formal-language/itu-t/x/x509/2019/AttributeCertificateDefinitions.html
[7] https://www.itu.int/ITU-T/formal-language/itu-t/x/x509/2019/AlgorithmObjectIdentifiers.html
[8] https://www.itu.int/ITU-T/formal-language/itu-t/x/x509/2019/AuthenticationFramework.html
[9] https://www.itu.int/ITU-T/formal-language/itu-t/x/x509/2019/CertificateExtensions.html
[10] https://www.itu.int/ITU-T/formal-language/itu-t/x/x511/2019/SpkmGssTokens.html
[11] https://www.itu.int/ITU-T/formal-language/itu-t/x/x511/2019/DirectoryAbstractService.html
[12] https://www.itu.int/ITU-T/formal-language/itu-t/x/x520/2019/PasswordPolicy.html
[13] https://www.itu.int/ITU-T/formal-language/itu-t/x/x520/2019/UpperBounds.html
[14] https://www.itu.int/ITU-T/formal-language/itu-t/x/x520/2019/SelectedAttributeTypes.html

На відміну від самого алгоритму КЕП, який визначено ДСТУ 4145, конверти визначаються не стандартами, а наказами міністерства юстиції: Проект наказу Адміністрації Держспецзв'язку та Держкомінфоматизації (2009) [15], Наказ Міністерства юстиції України 1236/5/453 [16]. Керуючись цими нормативними документами було створено файл KEP.asn1 [17], який є одним з трьох top-level файлів необхідниї для компіляції ASN.1 компілятором[18].

Існує небагато безкоштовних та повних компіляторів (генераторів парсерів) ASN.1 специфікацій. Erlang є прикладом системи, до складу якої входить першокласний безкоштовний з відкритою ліценцією ASN.1 компілятор, де файли в ASN.1 нотації можуть бути зкомпільовані безпосередньо Erlang компілятором:

```
> erlc AuthenticationFramework.asn1
> erlc InformationFramework.asn1
> erlc KEP.asn1
```

Створити файл підпису PKCS-7 можна за допомогою будь якої програми сертифікованої в Україні. Найпростіше отримати свою КЕП печатку будучи клієнтом ПриватБанку. За допомогою "Користувача ЦСК"компанії ІІТ ви можете підписувати файли використовуючи безкоштовну форму приватного ключа у вигляді звичайного файлу.

[15] http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article?art_id=77726
[16] https://zakon.rada.gov.ua/laws/show/z1401-12
[17] https://github.com/synrc/ca/blob/master/priv/kep/KEP.asn1
[18] https://asn1.erp.uno

10.1.1 Приклад використання

Щоб показати як користуватися КЕП, та прочитати атрибутивну інформацію з сертифікату, який вшитий в PCKS-7 повідомлення з криптографічним підписом, покажемо 5 функцій:

```
> CA.CAdES.readSignature
[
  {:certinfo, ~c"TINUA-2955020254",
   "СОХАЦЬКИЙ МАКСИМ ЕРОТЕЙОВИЧ",
   "МАКСИМ ЕРОТЕЙОВИЧ", "СОХАЦЬКИЙ",
   "СОХАЦЬКИЙ МАКСИМ ЕРОТЕЙОВИЧ",
   [
     subjectKeyIdentifier: "VNXfTvJQccGtPgNhUftIQZV+mUROTgroLotsbtYZsFE=",
     authorityKeyIdentifier: "XphNUm+C84/0vi5ABGgN/rOvysLkBHVNB9CuTISwfB0=",
     keyUsage: [<<6, 192>>],
     certificatePolicies: {"https://acsk.privatbank.ua/acskdoc",
      ["1.2.804.2.1.1.1.2.2", "1.3.6.1.5.5.7.2.1"]},
     basicConstraints: [],
     qcStatements: {"https://acsk.privatbank.ua",
      ["0.4.0.1862.1.1", "0.4.0.1862.1.5", "1.3.6.1.5.5.7.11.2",
       "0.4.0.194121.1.1", "1.2.804.2.1.1.1.2.1"]},
     cRLDistributionPoints: ["http://acsk.privatbank.ua/crl/PB-2023-S6.crl"],
     freshestCRL: ["http://acsk.privatbank.ua/crldelta/PB-Delta-2023-S6.crl"],
     authorityInfoAccess: [
       {"1.3.6.1.5.5.7.48.2",
        "http://acsk.privatbank.ua/arch/download/PB-2023.p7b"},
       {"1.3.6.1.5.5.7.48.1", "http://acsk.privatbank.ua/services/ocsp/"}
     ],
     subjectInfoAccess: [
       {"1.3.6.1.5.5.7.48.3", "http://acsk.privatbank.ua/services/tsp/"}
     ],
     subjectDirectoryAttributes: [
       {"1.2.804.2.1.1.1.11.1.4.7.1", "0"},
       {"1.2.804.2.1.1.1.11.1.4.1.1", "2955020254"}
     ]
   ], "ФІЗИЧНА ОСОБА", "", "", ~c"UA", "КИЇВ"},
  {:certinfo, ~c"UA-14360570-2310",
   "КНЕДП АЦСК АТ КБ ПРИВАТБАНК\"\"", "", "",
   "КНЕДП АЦСК АТ КБ ПРИВАТБАНК\"\"",
   [
     contentType: "0.6.9.42.840.113549.1.7.1",
     signingTime: "240221110356Z",
     messageDigest: "MfvlhoDVCPkptQRN+S2zNGp0nrOsS93mLdbcz/kZ9GI=",
     signingCertificateV2: 5400415814250126491315088041558718376138774192 68,
     contentTimestamp: {"1.2.840.113549.1.7.2",
      369952533463044024072847521187489 7026, "20240221110626Z",
      "MfvlhoDVCPkptQRN+S2zNGp0nrOsS93mLdbcz/kZ9GI="}
   ], "АТ КБ ПРИВАТБАНК\"\"", "", "", ~c"UA", "Київ"}
]
```

10.2 Криптографічні інформаційні повідомлення

Реалізації повинні підтримувати транспортування ключів, узгодження ключів і раніше розподілені симетричні ключі шифрування ключів, представлені ktri, kari та kekri відповідно.

Реалізації можуть підтримувати керування ключами на основі пароля, представлене pwri. Реалізації МОЖУТЬ підтримувати будь-які інші методи керування ключами, такі як шифрування на основі ідентифікації Боне-Франкліна та Боне-Бойєна (RFC 5409) або інші методи шифрування SYNRC, такі як варіанти KYBER Key Transport (LAMPS-WG) для постквантової криптографії (PQC).

IETF (SMIME-WG) стандарти: 5990, 5911, 5750–5754, 5652, 5408, 5409, 5275, 5126, 5035, 4853, 4490, 4262, 4134, 4056, 4010, 3850, 3851, 3852, 3854, 3855, 3657, 3560, 3565, 3537, 3394, 3369, 3370, 3274, 3114, 3278, 3218, 3211, 3217, 3183, 3185, 3125–3126, 3058, 2984, 2876, 2785, 2630, 2631, 2632 , 2633, 5083, 5084, 2634.

Сумісність: Erlang SSL, LibreSSL CMS, OpenSSL CMS, GnuPG S/MIME.

10.2.1 Головна функція

Специфікація синтаксису криптографічних повідомлень CMS X.509 для дисципліни RSA (Key Transport), ECC (Key Agreement), KEK (Key Encryption Key) для додатків Erlang/OTP, які ніколи не переживала heartbleed (!) CRYPTO та SSL. Реалізовано як модуль CMS програми CA.

```
defmodule CMS do
    def decrypt(cms, {schemeOID, privateKeyBin}) do
        {_,{:ContentInfo,_,{:EnvelopedData,_,_,x,y,_}}} = cms
        {:EncryptedContentInfo,_,{_,encOID,{_,<<_::16,iv::binary>>}},data} = y
              case :proplists.get_value(:kari,  x, []) do
        [] -> case :proplists.get_value(:ktri,  x, []) do
        [] -> case :proplists.get_value(:kekri, x, []) do
        [] -> case :proplists.get_value(:pwri,  x, []) do
        [] -> {:error, "Unknown Other Recepient Info"}
              pwri  -> pwri(pwri,   privateKeyBin, encOID, data, iv) end
              kekri -> kekri(kekri, privateKeyBin, encOID, data, iv) end
              ktri  -> ktri(ktri,   privateKeyBin, encOID, data, iv) end
              kari  -> kari(kari,   privateKeyBin, schemeOID, encOID, data, iv)
     end
    end
end
```

10.2.2 CMS-KARI-ECC

IETF 3278:2002 Використання алгоритмів криптографії еліптичних кривих (ECC) у синтаксисі криптографічних повідомлень (CMS) із підтримкою Suite B IETF 5008:2007, 6318:2011.

```
. . . openssl cms -encrypt -aes256 -in message.txt -out encrypted.txt \
                      -recip client.pem -keyopt ecdh_kdf_md:sha256
```

CMS Codec KARI: ECC+KDF/ECB+AES/KW+256/CBC:

```
def map(:'dhSinglePass-stdDH-sha512kdf-scheme'), do: :sha512
def map(:'dhSinglePass-stdDH-sha384kdf-scheme'), do: :sha384
def map(:'dhSinglePass-stdDH-sha256kdf-scheme'), do: :sha256
def eccCMS(ukm, bit), do:
    :'CMSECCAlgs-2009-02'.encode(:'ECC-CMS-SharedInfo', sharedInfo(ukm, bit))
def sharedInfo(ukm, len), do: {:'ECC-CMS-SharedInfo',
    {:'KeyWrapAlgorithm',{2,16,840,1,101,3,4,1,45},:asn1_NOVALUE}, ukm, <>}

def kari(kari, privateKeyBin, schemeOID, encOID, data, iv) do
    {_,:v3,{_,{_,_,publicKey}},ukm,{_,kdfOID,_},[{_,_,encryptedKey}]} = kari
    {scheme,_}  = CA.ALG.lookup(schemeOID)
    {kdf,_}     = CA.ALG.lookup(kdfOID)
    {enc,_}     = CA.ALG.lookup(encOID)
    sharedKey   = :crypto.compute_key(:ecdh,publicKey,privateKeyBin,scheme)
    {_,payload} = eccCMS(ukm, 256)
    derived     = KDF.derive(map(kdf), sharedKey, 32, payload)
    unwrap      = CA.AES.KW.unwrap(encryptedKey, derived)
    res         = CA.AES.decrypt(enc, data, unwrap, iv)
    {:ok, res}
end
def testDecryptECC(), do: CA.CMS.decrypt(testECC(), testPrivateKeyECC())

def testECC() do
    {:ok,base} = :file.read_file "priv/certs/encrypted.txt"
    [_,s] = :string.split base, "\n\n"
    x = :base64.decode s
    :'CryptographicMessageSyntax-2010'.decode(:ContentInfo, x)
end

def testPrivateKeyECC() do
    privateKey = :public_key.pem_entry_decode(pem("priv/certs/client.key"))
    {:'ECPrivateKey',_,privateKeyBin,{:namedCurve,schemeOID},_,_} = privateKey
    {schemeOID,privateKeyBin}
end
```

10.2.3 CMS-KEKRI-KEK

Інформація про одержувача ключа шифрування ключа, як визначено CMS IETF 5652:2009, 3852:2004, 3369:2002, 2630:1999.

```
07 01234567890123456789 256 . 2. openssl cms -decrypt -in encrypted2.txt -
    secretkeyid 07 \
            -secretkey 0123456789ABCDEF0123456789ABCDEF
```

CMS Codec KEKRI: KEK+AES-KW+CBC:

```
def kekri(kekri, privateKeyBin, encOID, data, iv) do
    {:'KEKRecipientInfo',_vsn,_,{_,kea,_},encryptedKey} = kekri
    _ = CA.ALG.lookup(kea)
    {enc,_} = CA.ALG.lookup(encOID)
    unwrap = CA.AES.KW.unwrap(encryptedKey,privateKeyBin)
    res = CA.AES.decrypt(enc, data, unwrap, iv)
    {:ok, res}
end
def testDecryptKEK(), do: CA.CMS.decrypt(testKEK(), testPrivateKeyKEK())

def testPrivateKeyKEK() do
    {:kek, :binary.decode_hex("0123456789ABCDEF0123456789ABCDEF")}
end

def testKEK() do
    {:ok,base} = :file.read_file "priv/certs/encrypted2.txt"
    [_,s] = :string.split base, "\n\n"
    x = :base64.decode s
    :'CryptographicMessageSyntax-2010'.decode(:ContentInfo, x)
end
```

10.2.4 CMS-KTRI-RSA

The very first CMS IETF 3852:1999:

```
  gpgsm --list-secret-keys
03878 ¿. gpgsm -u 0xD3C8F78A -d cms.bin
  1203878 ¿. openssl pkcs12 -in key.bin -nokeys -out public.pem
```

CMS Codec KTRI: RSA+RSAES-OAEP:

```
def ktri(ktri, privateKeyBin, encOID, data, iv) do
    {:'KeyTransRecipientInfo',_vsn,_,{_,schemeOID,_},key} = ktri
    {:rsaEncryption,_} = CA.ALG.lookup schemeOID
    {enc,_} = CA.ALG.lookup(encOID)
    sessionKey = :public_key.decrypt_private(key, privateKeyBin)
    res = CA.AES.decrypt(enc, data, sessionKey, iv)
    {:ok, res}
end
def testDecryptRSA(), do: CA.CMS.decrypt(testRSA(), testPrivateKeyRSA())

def testPrivateKeyRSA() do
    {:ok,bin} = :file.read_file("priv/rsa-cms.key")
    pki = :public_key.pem_decode(bin)
    [{:PrivateKeyInfo,_,_}] = pki
    rsa = :public_key.pem_entry_decode(hd(pki))
    {:'RSAPrivateKey',:'two-prime',_n,_e,_d,_,_,_,_,_,_} = rsa
    {:rsaEncryption,rsa}
end

def testRSA() do
    {:ok,x} = :file.read_file "priv/rsa-cms.bin"
    :'CryptographicMessageSyntax-2010'.decode(:ContentInfo, x)
end
```

10.2.5 KDF

KDF (MD5: 128, SHA: 160—512) and HKDF (HMAC) Key Derive functions used in ECC CMS schemes as of NIST SP 800-108r1.

```
defmodule KDF do
    def hl(:md5),    do: 16
    def hl(:sha),    do: 20
    def hl(:sha224), do: 28
    def hl(:sha256), do: 32
    def hl(:sha384), do: 48
    def hl(:sha512), do: 64

    def derive(h, d, len, x) do
        :binary.part(:lists.foldr(fn i, a ->
            :crypto.hash(h, d <> <> <> x) <> a
        end, <<>>, :lists.seq(1,round(Float.ceil(len/hl(h))))), 0, len)
    end
end
```

10.2.6 AES-KW

AES Key Wrap function is applicable to keys of 128/192/256 bit using AES-ECB encoding as of RFC 5649:2009 Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm.

```
-define(MSB64,      1/unsigned-big-integer-unit:64).
-define(DEFAULT_IV, << 16#A6A6A6A6A6A6A6A6:?MSB64 >>).

unwrap(CipherText, KEK) -> unwrap(CipherText, KEK, ?DEFAULT_IV).
unwrap(CipherText, KEK, IV)
        when (byte_size(CipherText) rem 8) =:= 0
        andalso (bit_size(KEK) =:= 128
            orelse bit_size(KEK) =:= 192
            orelse bit_size(KEK) =:= 256) ->
    BlockCount = (byte_size(CipherText) div 8) - 1,
    IVSize = byte_size(IV),
    case do_unwrap(CipherText, 5, BlockCount, KEK) of
        << IV:IVSize/binary, PlainText/binary >> ->
            PlainText;
        _ ->
            erlang:error({badarg, [CipherText, KEK, IV]})
    end.

codec(128) -> aes_128_ecb;
codec(192) -> aes_192_ecb;
codec(256) -> aes_256_ecb.

do_unwrap(Buffer, J, _BlockCount, _KEK) when J < 0 -> Buffer;
do_unwrap(Buffer, J, BlockCount, KEK) ->
    do_unwrap(do_unwrap(Buffer, J, BlockCount, BlockCount, KEK),
              J - 1, BlockCount, KEK).
do_unwrap(Buffer, _J, I, _BlockCount, _KEK) when I < 1 -> Buffer;
do_unwrap(<< A0:?MSB64, Rest/binary >>, J, I, BlockCount, KEK) ->
    HeadSize = (I - 1) * 8,
    << Head:HeadSize/binary, B0:8/binary, Tail/binary >> = Rest,
    Round = (BlockCount * J) + I,
    A1 = A0 bxor Round,
    Data = << A1:?MSB64, B0/binary >>,
    << A2:8/binary, B1/binary >>
        = crypto:crypto_one_time(codec(bit_size(KEK)),
          KEK, ?DEFAULT_IV, Data, [{encrypt,false}]),
    do_unwrap(<< A2/binary, Head/binary, B1/binary,
          Tail/binary >>, J, I - 1, BlockCount, KEK).
```

10.2.7 AES-256

All AES-256 flavours are implemented for a wide range of ECC Key Agreement schemes.

```
def decrypt(crypto_codec, data, key, iv \\ :crypto.strong_rand_bytes(16))
def decrypt(:'id-aes256-ECB',data,key,iv), do: decryptAES256ECB(data,key,iv)
def decrypt(:'id-aes256-CBC',data,key,iv), do: decryptAES256CBC(data,key,iv)
def decrypt(:'id-aes256-GCM',data,key,iv), do: decryptAES256GCM(data,key,iv)
def decrypt(:'id-aes256-CCM',data,key,iv), do: decryptAES256CCM(data,key,iv)
def test() do
  [
    check_SECP384R1_GCM256(),
    check_X25519_GCM256(),
    check_C2PNB368w1_GCM256(),
    check_BrainPoolP512t1_GCM256(),
    check_BrainPoolP512t1_GCM256(),
    check_SECT571_GCM256(),
    check_X448_GCM256(),
    check_X448_CBC256(),
    check_X448_ECB256(),
  ]
end
```

10.3 Імплементація CMP сервера у складі АЦСК

IETF follow up (PKIX): 7030, 6960, 6818, 6844, 6712, 6664, 6402, 6277, 6170, 6024, 6025, 5934, 5912–5914, 5877, 5816, 5755, 5756, 5758, 5697, 5636, 5480, 5272–5274, 5280, 5055, 5019, 4985, 4683, 4630, 4476, 4387, 4325, 4158, 4210, 4211, 4055, 4043, 3874, 3779, 3820, 3739, 3709, 3628, 3161, 3029, 2797, 2559, 2587, 3039, 3029, 2511, 2510.

Compatibility: OpenSSL, Cisco, Red Hat, Siemens, Nokia, IBM.

Ця стаття могла би називати «Як написати CMP сервер за 30 хвилин», але на відміну від попередньої статті про LDAP, ця вже покриває більше ніж тузінь ASN.1 файлів, добре шо ми вже познайомилися з CMS та LDAP бібіліотеками та їх ASN.1 файлами. В цій статті про CMP нас в основному цікавитимуть PKIXCMP-2009, PKIXCRMF-2009 та EnrollmentMessageSyntax-2009 для CMC.

```
CMS-AES-CCM-and-AES-GCM-2009.asn1
CMSAesRsaesOaep-2009.asn1
CMSECCAlgs-2009-02.asn1
CMSECDHAlgs-2017.asn1
CryptographicMessageSyntax-2009.asn1
CryptographicMessageSyntax-2010.asn1
CryptographicMessageSyntaxAlgorithms-2009.asn1
EnrollmentMessageSyntax-2009.asn1
PKCS-10.asn1
PKCS-7.asn1
PKIX1Explicit-2009.asn1
PKIX1Implicit-2009.asn1
PKIXAlgs-2009.asn1
PKIXCMP-2009.asn1
PKIXCRMF-2009.asn1
```

10.3.1 CSR

Отже починається написання CMP серверу з найголовнішої його функції: видачі сертифікату по PCKS-10 CSR реквесту. Схема наступна: Клієнт генерує приватний ключ, конвертує його в PEM файл, відсилає як P10CR повідомлення у складі payload PKIMessage, отримує відповідь CP, після чого клієнт шле ще одне повідомлення CERTCONF, після якого CMP сервер повинен відповисти PKICONF повідомленням.

```
def csr(user) do
    {ca_key, ca} = read_ca()
    priv = X509.PrivateKey.new_ec(:secp384r1)
    der = :public_key.der_encode(:ECPrivateKey, priv)
    pem = :public_key.pem_encode([{:ECPrivateKey, der, :not_encrypted}])
    :file.write_file(user <> ".key", pem)
    :io.format '~p~n', [priv]
    csr = X509.CSR.new(priv, "/C=UA/L=Kyiv/O=SYNRC/CN=" <> user,
       extension_request: [
          X509.Certificate.Extension.subject_alt_name(["n2o.dev"])])
    :io.format 'CSR: ~p~n', [csr]
    :file.write_file(user <> ".csr", X509.CSR.to_pem(csr))
    true = X509.CSR.valid?(csr)
    subject = X509.CSR.subject(csr)
    :io.format 'Subject ~p~n', [subject]
    :io.format 'CSR ~p~n', [csr]
    X509.Certificate.new(X509.CSR.public_key(csr), subject, ca, ca_key,
       extensions: [subject_alt_name:
         X509.Certificate.Extension.subject_alt_name(["n2o.dev", "erp.uno"])
    ])
    csr
end
```

Перед початком роботи CMP сервера повинен бути згенерований рутовий CA сертифікат з приватним ключем, ці два файла ми зберігаємо на диск, і у всіх подальших операціях користуємося ними. Для генерації файлів використовуємо функцію CA.CSR.ca.

```
def ca() do
    ca_key = X509.PrivateKey.new_ec(:secp384r1)
    ca = X509.Certificate.self_signed(ca_key,
          "/C=UA/L=Kyiv/O=SYNRC/CN=CSR-CMP", template: :root_ca)
    der = :public_key.der_encode(:ECPrivateKey, ca_key)
    pem = :public_key.pem_encode([{:ECPrivateKey, der, :not_encrypted}])
    :file.write_file "ca.key", pem
    :file.write_file "ca.pem", X509.Certificate.to_pem(ca)
    {ca_key, ca}
end

def read_ca() do
    {:ok, ca_key_bin} = :file.read_file "ca.key"
    {:ok, ca_bin} = :file.read_file "ca.pem"
    {:ok, ca_key} = X509.PrivateKey.from_pem ca_key_bin
    {:ok, ca} = X509.Certificate.from_pem ca_bin
    {ca_key, ca}
end
```

Для одноразовоїї генерації серверних сертифікатів які обсуговують клієнтські TLS сесії можна використати наступний код.

```
def server(name) do
    {ca_key, ca} = read_ca()
    server_key = X509.PrivateKey.new_ec(:secp384r1)
      X509.Certificate.new(X509.PublicKey.derive(server_key),
         "/C=UA/L=Kyiv/O=SYNRC/CN=" <> name, ca, ca_key,
         extensions: [subject_alt_name:
            X509.Certificate.Extension.subject_alt_name(["n2o.dev", "erp.uno"
    ]) ])
end
```

10.3.2 CMS

Детально сімейство протоколів і CMS кодування описано в окремій статті присвяченій CMS Compliance. CMS кодування використовується тільки для CMC сервера, тому ми це поки висвітлювати не будемо.

10.3.2.1 CMP/CSR/TCP

RFC 6712, 4210. Для початку напишемо простий PKIMessage сервер.

```
defmodule CA.CMP do
  @moduledoc "CA/CMP TCP server."
  require CA

  def start(), do: :erlang.spawn(fn -> listen(1829) end)
  def listen(port) do
      {:ok, socket} = :gen_tcp.listen(port,
        [:binary, {:packet, 0}, {:active, false}, {:reuseaddr, true}])
      accept(socket)
  end

  def accept(socket) do
      {:ok, fd} = :gen_tcp.accept(socket)
      :erlang.spawn(fn -> __MODULE__.loop(fd) end)
      accept(socket)
  end

  def loop(socket) do
      case :gen_tcp.recv(socket, 0) do
           {:ok, data} ->
               [headers,body] = :string.split data, "\r\n\r\n", :all
               {:ok,dec} = :'PKIXCMP-2009'.decode(:'PKIMessage', body)
               {:PKIMessage, header, body, code, _extra} = dec
               __MODULE__.message(socket, header, body, code)
               loop(socket)
          {:error, :closed} -> :exit
      end
  end
```

10.3.2.2 PKIMessage.protection

Розберемося з полем PKIMessage.protection, в якому зберігається результат PBKDF2 алгоритма. Майте на увазі шо OpenSSL за замовчування використовує 20-байтні ключі та HMAC/SHA-1 у якості MAC функції, хоча OWF в 500 ітераціях обчислюється за допомогою OWF функції SHA-256.

10.3.2.3 ANSWER

Оскільки CMP сервер повинен працювати по HTTP/1.0 згідно стандартів додаємо необхідні HTTP заголовки.

```
def answer(socket, header, body, code) do
     message = CA."PKIMessage"(header: header, body: body, protection: code)
     {:ok, bytes} = :'PKIXCMP-2009'.encode(:'PKIMessage', message)
     res =  "HTTP/1.0 200 OK\r\n"
         <> "Server: SYNRC CA/CMP\r\n"
         <> "Content-Type: application/pkixcmp\r\n\r\n"
         <> :erlang.iolist_to_binary(bytes)
     :gen_tcp.send(socket, res)
 end
```

10.3.2.4 P10CR/CP

Запускаємо сервер та генеруємо сертифікати CA та CSR користувача:

```
. iex -S mix
> CA.CSR.ca
> CA.CSR.csr "maxim"
```

Запускаємо клієнтський запит за допомогою OpenSSL:

```
# openssl cmp -cmd p10cr -server localhost:1829 \
#             -path . -srvcert ca.pem -ref cmptestp10cr \
#             -secret pass:0000 -certout . client.csr
```

Пишемо функцію видачі сертифікату:

```
def message(socket, header, {:p10cr, csr} = body, code) do
      {:PKIHeader, pvno, from, to, messageTime, protectionAlg,
         _senderKID, _recipKID, transactionID, senderNonce,
         _recipNonce, _freeText, _generalInfo} = header
      true = code == validateProtection(header, body, code)

      {ca_key, ca} = CA.CSR.read_ca()
      subject = X509.CSR.subject(csr)
      :io.format '~p~n',[subject]
      true = X509.CSR.valid?(CA.parseSubj(csr))
      cert = X509.Certificate.new(X509.CSR.public_key(csr),
         CA.CAdES.subj(subject), ca, ca_key,
         extensions: [subject_alt_name:
            X509.Certificate.Extension.subject_alt_name(["synrc.com"]) ])

      reply = CA."CertRepMessage"(response:
            [ CA."CertResponse"(certReqId: 0,
              certifiedKeyPair: CA."CertifiedKeyPair"(certOrEncCert:
                {:certificate, {:x509v3PKCert, CA.convertOTPtoPKIX(cert)}}),
              status: CA."PKIStatusInfo"(status: 0))])

      pkibody = {:cp, reply}
      pkiheader = CA."PKIHeader"(sender: to, recipient: from, pvno: pvno,
          recipNonce: senderNonce, transactionID: transactionID,
          protectionAlg: protectionAlg, messageTime: messageTime)
      answer(socket, pkiheader, pkibody,
          validateProtection(pkiheader, pkibody, code))
  end
```

10.3.2.5 CERTCONF/PKICONF

```
def message(socket, header, {:certConf, statuses}, code) do
      {:PKIHeader, _, from, to, _, _, _, _, _, senderNonce, _, _, _} = header

      :lists.map(fn {:CertStatus,bin,no,{:PKIStatusInfo, :accepted, _, _}} ->
          :logger.info 'CERTCONF ~p request ~p~n', [no,:binary.part(bin,0,8)]
      end, statuses)

      pkibody = {:pkiconf, :asn1_NOVALUE}
      pkiheader = CA."PKIHeader"(header, sender: to, recipient: from,
          recipNonce: senderNonce)
      answer(socket, pkiheader, pkibody,
          validateProtection(pkiheader, pkibody, code))
  end
```

В результаті в консолі повинні спостерігати:

```
CMP info: sending P10CR
CMP info: received CP
CMP info: sending CERTCONF
CMP info: received PKICONF
CMP info: received 1 enrolled certificate(s), saving to file 'maxim.pem'
```

10.3.2.6 GENM/GENP

Далі можете написати інші функції:

```
# openssl cmp -cmd genm -server 127.0.0.1:1829 \
#             -recipient "/CN=CMPserver" -ref 1234 -secret pass:0000


def message(_socket, _header, {:genm, req} = _body, _code) do
      :io.format 'generalMessage: ~p~n', [req]
  end
```

10.3.2.7 IR/IP

```
# openssl cmp -cmd ir -server 127.0.0.1:1829 \
#             -path . -srvcert ca.pem -ref NewUser \
#             -secret pass:0000 -certout maxim.pem \
#             -newkey maxim.key -subject "/CN=maxim/O=SYNRC/ST=Kyiv/C=UA"
```

```
 def message(_socket, _header, {:ir, req}, _) do
      :lists.map(fn {:CertReqMsg, req, sig, code} ->
         :io.format 'request: ~p~n', [req]
         :io.format 'signature: ~p~n', [sig]
         :io.format 'code: ~p~n', [code]
      end, req)
  end
```

10.3.2.8 CR/CP

```
# openssl cmp -cmd cr -server 127.0.0.1:1829 \
#             -path . -srvcert ca.pem -ref NewUser \
#             -secret pass:0000 -certout maxim.pem \
#             -newkey maxim.key -subject "/CN=maxim/O=SYNRC/ST=Kyiv/C=UA"
```

10.3.2.9 Висновки

```
defmodule CA do
  use Application
  use Supervisor

  require Record
  Enum.each(Record.extract_all(from_lib: "ca/include/PKIXCMP-2009.hrl"),
            fn {name, definition} -> Record.defrecord(name, definition) end)

  Enum.each(Record.extract_all(from_lib: "public_key/include/public_key.hrl"),
            fn {name, definition} -> Record.defrecord(name, definition) end)

  def init([]), do: {:ok, { {:one_for_one, 5, 10}, []} }
  def start(_type, _args) do
      :logger.add_handlers(:ldap)
      CA.CMP.start
      CA.CMC.start
      :supervisor.start_link({:local, __MODULE__}, __MODULE__, [])
  end
```

Ну PasswordBasedMac в нас є, тепер треба DHMac, але shared secret можна і в PBM засунути. Є ше Proof Of Posession (POP) — там зразу ECDSA verify. Я до речі думаю в CA тримати ключі для всіх кривих, і коли я виставлятиму сервіс то я буду виставляти його на N портах і N ключах, щоб будь який клієнтський TLS сертифікат приймався як рідний! Chat X.509 дає можливість вибирати TLS сертифікати автоматично по обраних кривих. Ви вибираєте під якими ключами сьогодні заходити. LDAP, MQTT, NS, CA — в кожного сервісу свої N портів і N серверних TLS сертифікатів. Передбачається що перший сертифікат видається DH по TCP а потім зразу всьо переходить в TLS режим і всі наступні сертифікати вже видаються всередині клієнтського TLS. При реєстрації користувач зразу доступний в LDAP якшо захотів зробити себе відкритим для пошуку. Після реєстрації пошук в директорії і френдування (обмін ключами) і поїхали чат в обгортках CAdES, CMS, ECDSA/AES — лейби біля повідомлень ПІДПИС/ШИФР.

10.4 Центри сертифікації CA, АЦСК, ЦЗО та ОЗО

10.5 Безпечна система доменних імен DNSSEC

10.6 Система директорії підприємства LDAP

10.7 Протокол розмежування доступу ABAC

Розділ 11

Апробація

ЄРЗ (Єдиний реєстр зброї НПУ), СУСЗЦЗ (Система управління силами та засобами цивільного захисту ДСНС), ЄІС (Єдина інформаційна система МВС), ФП МТРЗ (Функціональна підсистема матеріально-технічного та ресурсного забезпечення МВС), ГСЦ (Головний сервісний центр МВС).

Розділ 12

Висновки

ЄРЗ (Єдиний реєстр зброї НПУ), СУСЗЦЗ (Система управління силами та засобами цивільного захисту ДСНС), ЄІС (Єдина інформаційна система МВС), ФП МТРЗ (Функціональна підсистема матеріально-технічного та ресурсного забезпечення МВС), ГСЦ (Головний сервісний центр МВС).

Список використаних джерел