



ПАТЕНТНО ВЕДОМСТВО  
на Република България

# СВИДЕТЕЛСТВО

ЗА  
РЕГИСТРАЦИЯ НА ПОЛЕЗЕН МОДЕЛ

**Рег. № 5104 U1**

**Заявка № 6432**

**Дата на заявяване:** 17/04/2025

**Приоритет:**

**Срок на действие:** 17/04/2029

**Публ. за регистрация:** 29/08/2025

**Наименование:** КОМПЮТЪРНО  
РЕАЛИЗИРАНА СИСТЕМА ЗА  
ОБРАБОТКА НА ГОЛЕМИ ДАННИ

**Притежател/и:**

Университет по библиотекознание и  
информационни технологии UNIBIT  
бул. Цариградско шосе 119,  
1784 София [BG]

Институт по информационни и  
комуникационни технологии - БАН  
ул. "Акад. Георги Бончев", бл. 2  
1336 София [BG]

**Изобретател/и:**

Александър Иванов Шиќаланов  
Леонид Михайлов Кирилов  
Евгения Петрова Ковачева  
Румен Василев Николов  
Елена Дикова Шойкова-Стоянова  
Александър Илиев Илиев  
Любомир Иванов Гоцев

ОЛЯ ДИМИТРОВА  
ПРЕДСЕДАТЕЛ



ОПИСАНИЕ КЪМ СВИДЕТЕЛСТВО  
ЗА РЕГИСТРАЦИЯ  
НА ПОЛЕЗЕН МОДЕЛ

(51) Int. Cl.  
G 06 F 15/00 (2006.01)  
G 06 Q 50/00 (2024.01)  
G 06 F 16/00 (2019.01)  
(52) CPC  
G 06 F 15/00 (2013.01)  
G 06 Q 50/00 (2013.01)  
G 06 F 16/00 (2019.01)

## ПАТЕНТНО ВЕДОМСТВО

(21) Заявителски № 6432  
(22) Заявено на 17.04.2025  
(24) Начало на действие  
на регистрацията от: 17.04.2025

**Приоритетни данни**

(45) Отпечатано на 29.08.2025  
(46) Публикувано в  
бюлетин № 202508.2 на 29.08.2025  
(56) Информационни източници:  
(62) Разделена заявка от рег. №  
(66) Трансформирано от:  
(67) Паралелно на:

(73) Притежатели (и):

**УНИВЕРСИТЕТ ПО БИБЛИОТЕКОЗНАНИЕ И  
ИНФОРМАЦИОННИ ТЕХНОЛОГИИ, UNIBIT  
1784 СОФИЯ, БУЛ. ЦАРИГРАДСКО ШОСЕ 119  
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И  
КОМУНИКАЦИОННИ ТЕХНОЛОГИИ - БАН  
1336 СОФИЯ, УЛ. АКАД."ГЕОРГИ БОНЧЕВ", БЛ. 2**

(72) Изобретател(и):

**Александър Иванов Шикаланов  
Леонид Михайлов Кирилов  
Евгения Петрова Ковачева  
Румен Василев Николов  
Елена Дикова Шойкова-Стоянова  
Александър Илиев Илиев  
Любомир Иванов Гоцев**

(74) Представител по индустриална собственост:

**Константин Неделчев Тахтаджиев  
1404 София, ж. к. Гоце Делчев бл.237, вх. Б, ет. 4,  
ап. 28**

(86) № на РСТ заявка:

(87) № и дата на РСТ публикация:

**(54) КОМПЮТЪРНО РЕАЛИЗИРАНА  
СИСТЕМА ЗА ОБРАБОТКА НА ГОЛЕМИ  
ДАННИ**

(57) Компютърно реализирана система за обработка на големи данни Big Data, включваща: първи слой (10) за получаване на данни от източници на данни; втори слой (20) за обработка и съхранение на данни, включващ обработващ подслой (22) и съхраняващ подслой (24); трети слой (30) за приложения, използващи обработените данни; характеризираща се с

това, че системата е конфигурируема чрез софтуерен-контрол да работи в: трислоен режим, при който обработващият подслой (22) и съхраняващият подслой (24) работят независимо и свързано; и двуслоен режим, при който обработващият подслой (22) и съхраняващият подслой (24) е условно изключен.

12 претенции 7 фигури



## (54) КОМПЮТЪРНО РЕАЛИЗИРАНА СИСТЕМА ЗА ОБРАБОТКА НА ГОЛЕМИ ДАННИ

### Област на техниката

Настоящият полезен модел се отнася до областта на компютърните системи, и по-конкретно до компютърно реализирани системи за обработка на големи данни (Big Data).

### Предшестващо състояние на техниката

През последните години областта на обработката на големи масиви от данни, позната като Big Data, претърпява бързо и динамично развитие. Това се дължи на няколко фактора, включително напредъка в хардуерните и софтуерните инструменти, създаването на мощни алгоритми за обработка на данни и масовата дигитализация на съвременното общество. Тези тенденции позволяват формулирането и решаването на много нови проблеми, които изискват работа с огромни обеми от данни, поставяйки на дневен ред въпроси относно тяхното съхранение, обработка, анализ, визуализация и сигурност [1 - 5].

Big Data се характеризира като колекция от голям обем (Volume) данни, висока скорост на генериране и обработка на данни (Velocity) и голямо разнообразие от източници, типове и структури на данните (Variety). Допълнителни характеристики включват достоверност на данните (Veracity), ценност (Value), взаимосвързаност (Valency), изменчивост (Volatility) и валидност (Validity) [6 - 7]. Тези особености създават комплексни предизвикателства при управлението и анализирането на данните.

Един от ключовите въпроси в областта е как да се организират хардуерните компоненти и софтуерните инструменти, за да се осигури ефективна работа с Big Data за различни цели [8 - 9].

До момента са известни няколко типа Big Data архитектури, в това число и теоретични.

В [8] - Tejada, Z. (2022) Big Data architectures. Microsoft, <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/> е представен обобщен модел (теоретичен) на Big Data архитектура - както е показано на фигура 1. Състои се от следните компоненти:

- Източници на данни - един или повече;
- Хранилище на данни - обикновено това е разпределено файлово хранилище (data lake);
- Пакетна обработка - batch processing;
- Приемане на съобщения в реално време - real-time message ingestion (за източници на данни в реално време) - обикновено хранилище на данни или хранилище/буфер за съхранение на съобщения;
- Поточна обработка - stream processing - обработка на съобщенията, постъпващи в реално време (филтриране, обобщаване и т. н.) и подготовка на данните за анализ. Като резултат, поточните данни са поставени в изходящо хранилище;
- Хранилище на аналитични данни - analytical data store - подготвя данните за анализ (структуриран формат) и след това се използват аналитични инструменти;
- Анализ и докладване - analytics and reporting - представяне на резултатите на крайните потребители. За тази цел е възможно да се използва отделен слой за моделиране на данни с цел подходяща визуализация;
- Оркестрация (организация, оркестриране) - orchestration - за автоматизиране на повтарящи се операции във формат на работни потоци (трансформация на входящите данни, преместване на данни, обработка, изчисления, анализ, визуализация и др.).

Съществуват различни архитектури за обработка на големи данни, включително теоретични модели [8 - 10] и практически реализации като Lambda архитектурата, Карпа архитектурата, архитектури за Интернет на нещата (IoT) и микросървисни архитектури. Всяка от тези архитектури има свои особености и е предназначена да адресира специфични аспекти на работата с големи данни.

Например, Lambda архитектурата комбинира пакетна и поточна обработка на данни, като се стреми да предостави висока точност и актуалност на данните. Тя обаче може да бъде сложна за имплементиране и поддръжка. Lambda архитектурата (LA), показана на фигура 2. е предложена за първи път от Nathan Marz - [11]. Базира се на три слоя: "партиден слой - batch layer" (разпределена файлова система) за съхраняване на всички данни; "скоростен/бърз слой - speed layer" за обработка на данни между две партии в реално време и "обслужващ слой" за индексване на партидите за по-добро управление на заявките - querying. LA е проектирана да минимизира забавянето на актуализациите и да увеличи максимално точността, което се дължи на партидният слой.

Карпа архитектурата е разработена с цел опростяване на обработката, като се фокусира върху поточната обработка, но това може да доведе до компромиси в точността при определени сценарии. Карпа архитектурата (KA), показана на фигура 3, е разработена от Jay Kreps през 2014 г. като алтернатива на Lambda архитектурата, с цел опростяване на структурата - [12]. KA се базира на два слоя: "обслужващ слой - serving layer" за всички входящи данни и "скоростен слой - speed layer", допълнен с функция за повторна обработка. Една от организациите, използващи KA, е LinkedIn, основана от J. Kreps. Сравнение между двете архитектури LA и KA е направено в [13]. Резултатите показват по-добра точност за LA с 9% за сметка на 2,2 пъти по-голямо време за обработка и 10 - 20% повече натоварване на процесорите - CPU.

Архитектура на Интернет-на-нещата (IoTА). Днес са известни редица IoT архитектури - [8, 14]. На фигура 4. е показан обобщен теоретичен модел на архитектура на Интернет-на-нещата съгл. [13]. Архитектурите от тип Интернет-на-нещата са събитийно ориентирани и могат да имат огромен брой приложения и реализации навсякъде, където се използват интелигентни устройства, като интелигентен дом, интелигентен град, мониторинг, сигурност и т. н.

Микросървисните архитектури (MSA) предоставят възможност за мащабируемост и гъвкавост, но изискват внимателно управление на комуникацията между отделните услуги. Такава архитектура е ефективна, когато трябва да се изпълнят голям брой услуги, които са относително слабо свързани помежду си. Всяка (микро) услуга се обработва отделно и независимо - [15]. Архитектурата тип микросървис произлиза от известната архитектура, ориентирана към услуги (SOA). Трябва да се отбележи, че двете архитектури често се обсъждат и коментират съвместно и даже се противопоставят една на друга без никаква причина. Архитектурата тип микросървис се използва в дейностите на големи организации като Amazon, eBay, Netflix.

Архитектурата Zeta (ZA). ZA е предложен от Джим Скот - [16]. Това е архитектура, много тясно свързана с дейността на една организация. Всяка дейност се поставя в отделен контейнер, където приложенията могат да се изпълняват отделно.

По този начин могат да се поддържат голям брой (различни) дейности. Като пример за приложение на Zeta архитектура е услугата Gmail на Google.

Допълнително, интегрирането на различни типове данни от разнообразни източници представлява значително предизвикателство. Необходимо е да се осигури технологична независимост и възможност за лесно интегриране със съществуващи и бъдещи системи. Това включва управление на данни от интелигентни устройства и сензорни мрежи, както и осигуряване на сигурността на системната архитектура при обмена на информация.

С нарастващия брой приложения на Интернет на нещата, възниква необходимост от решения, които позволяват ефективна работа с интелигентни устройства и осигуряват лесна комуникация с външни устройства. Това поставя акцент върху разработването на отворени системни архитектури, които могат да се адаптират към постоянно променящите се изисквания и технологии.

Въпреки съществуващите решения и архитектури в областта на Big Data, продължават да съществуват предизвикателства, които изискват допълнителни усилия и новаторски подходи. Необходима е по-голяма гъвкавост при обработката на данни, подобрена интеграция между хардуерните и софтуерните компоненти и по-ефективни методи за управление на сигурността и комуникацията с разнообразни устройства.

Следователно, има необходимост от допълнителни усъвършенствания в тази област, за да се адресират поне част от тези предизвикателства.

#### **Техническа същност на полезния модел**

Целта на полезния модел е да предостави компютърно реализирана система за обработка на големи данни Big Data, която реализира нова хибридна архитектура, позволяваща софтуерно превключване между трислойна и двуслойна структура. Това осигурява възможност за баланс между висока точност и висока бързина при работа с Big Data, като по този начин се решава проблема с адаптиране на архитектурата към конкретните нужди на различните приложения. Тази цел се постига чрез аспектите на настоящия полезен модел.

В първия аспект, настоящият полезен модел се отнася до компютърно реализирана система за обработка на големи данни Big Data, включваща: първи слой за получаване на данни от източници на данни; втори слой, свързан с първия слой, за обработка и съхранение на данни, включващ обработващ подслон и съхраняващ подслон, свързан с обработващия подслон; трети слой, свързан с втория слой, за осигуряване на достъп на потребителски приложения, използващи обработените данни; характеризира се с това, че компютърно реализираната система е конфигурируема чрез настройки (инициализация на софтуера) да работи в трислоен или двуслоен режим.

В едно предпочитано примерно изпълнение, първият слой може да включва:

IoT устройство; IoT агент; генеричен активатор; контекстен брокер за управление на контекстната информация; компонент за обработка на контекстна информация според предварително създадени правила, при което IoT устройството е свързано към IoT агента; IoT агентът е свързан към генеричния активатор; генеричният активатор е свързан към контекстния брокер за управление на контекстната информация; контекстният брокер за управление на контекстната информация е свързан към компонента за обработка на контекстна информация според предварително създадени правила. Това позволява ефективно събиране и управление на данни от различни IoT източници.

В едно предпочитано примерно изпълнение, компонентът за обработка на контекстна информация може да е PROTON генеричен активатор, IoT агентът за прехвърляне на данни между HTTP адаптера и контекстния брокер може да е IDAS, а контекстният брокер за управление на контекстната информация може да е Orion. Използването на тези утвърдени компоненти осигурява надеждност и съвместимост.

В едно предпочитано примерно изпълнение, контекстният брокер може да е конфигуриран да моделира контекстната информация като NGSI-LD обекти. Това позволява стандартизирано представяне на контекстните данни.

В едно предпочитано примерно изпълнение, обработващият подслой може да е генеричен активатор COSMOS GE и да включва: Apache Spark за широкомащабна обработка на данни, свързан с контекстния брокер чрез първи конектор; Apache Flink за машинно обучение, свързан с контекстния брокер чрез втори конектор. Използването на тези мощни инструменти за анализ на големи данни осигурява висока производителност и гъвкавост.

В едно предпочитано примерно изпълнение, съхраняващият подслой може да включва: компонент за трайно съхранение на данни от контекстния брокер, свързан към контекстния брокер 19 и към реляционна база данни и/или NOSQL база данни и към компонент за публикуване, търсене и откриване на отворени набори от данни. Това позволява надеждно съхранение и лесен достъп до данните.

В едно предпочитано примерно изпълнение, компонентът за трайно съхранение може да е Draco генеричен активатор, реляционните бази данни може да са MySQL, NOSQL базите данни може да са Mongo, а компонентът за публикуване, търсене и откриване на отворени набори от данни може да е SKAN разширение. Използването на тези утвърдени технологии гарантира стабилност и съвместимост.

В едно предпочитано примерно изпълнение, третият слой може да включва: приложения, свързани директно с базите данни или с компонентите на втория слой; компонент, свързан с приложенията, за улесняване разработването на персонализирани оперативни табла за управление. Това позволява лесно изграждане на потребителски приложения, използващи обработените големи данни.

В едно предпочитано примерно изпълнение, компонентът за улесняване разработването на персонализирани оперативни табла може да е Wirecloud генеричен активатор. Това осигурява мощен и гъвкав инструмент за визуализация.

В едно предпочитано примерно изпълнение, системата може допълнително да включва компонент за обработка на контекстна информация според предварително създадени правила. Това позволява автоматизирана обработка на данните въз основа на дефинирана бизнес логика.

В едно предпочитано примерно изпълнение, системата може допълнително да включва високопроизводителен сървър, свързан със първия, втория и третия слой, с: поне два многоядрени процесора; поне два графични процесора; дисков масив за съхранение на големи данни. Използването на мощен хардуер осигурява висока производителност при обработката на големи обеми данни.

Преимущество на варианти на настоящия полезен модел е, че системата може да обработва различни типове данни. Преимущество на варианти на настоящия полезен модел е, че системата може да работи с разнообразни източници на данни. Преимущество на варианти на настоящия полезен модел е, че архитектурата на системата е сигурна. Преимущество на варианти на настоящия полезен модел е, че системата позволява интеграция със съществуващи и бъдещи системи без да се засяга тяхната архитектура.

Преимущество на варианти на настоящия полезен модел е, че системата е технологично независима. Преимущество на варианти на настоящия полезен модел е, че системата може да работи с интелигентни устройства. Преимущество на варианти на настоящия полезен модел е, че системата осигурява лесна комуникация с външни устройства, включително интелигентни устройства. Преимущество на варианти на настоящия полезен модел е, че архитектурата на системата е от отворен тип. Преимущество на варианти на настоящия полезен модел е, че хибридната архитектура съчетава предимствата на Lambda и Карра архитектурите в едно, като позволява превключване между трислоен и двуслоен режим за постигане на висока точност или бързина при необходимост. Преимущество на варианти на настоящия полезен модел е, че системата е изградена на базата на компоненти с отворен код. Преимущество на варианти на настоящия полезен модел е, че системата отговаря както на изискванията за бързина, така и за точност, чрез софтуерно конфигурируемо превключване между режими. Преимущество на варианти на настоящия полезен модел е, че използването на FIWARE компоненти позволява изграждането на напълно многофункционална Big Data архитектура. Преимущество на варианти на настоящия полезен модел е, че системата може да се прилага в различни области като интелигентни градове, управление на трафика, обществена администрация и други. Преимущество на варианти на настоящия полезен модел е, че скалируемият и гъвкав дизайн на системата позволява ефективна обработка и съхранение на големи данни. Преимущество на варианти на настоящия полезен модел е, че се постига оптимално натоварване на хардуера и съвместна работа на отделните компоненти. Преимущество на варианти на настоящия полезен модел е, че наличието на високопроизводителни графични процесори ускорява изчисленията, свързани с анализа на големи данни. Преимущество на варианти на настоящия полезен модел е, че системата може да обработва и анализира данни както в пакетен, така и в потоков режим. Преимущество на варианти на настоящия полезен модел е, че системата поддържа изискванията за работа с Big Data като съхранение, обработка, анализ, изчисление, визуализация, сигурност и други. Преимущество на варианти на настоящия полезен модел е, че системата осигурява гъвкавост чрез възможност за интегриране със съществуващи или бъдещи системи без да се засяга тяхната архитектура. Преимущество на варианти на настоящия полезен модел е, че системата може софтуерно да се превключва между режими в зависимост от изискванията на приложенията за бързина или точност при обработката на големи данни.

Особени и предпочитани аспекти на полезния модел са изложени в приложените независими и зависими претенции. Характеристики от зависимите претенции могат да бъдат комбинирани с характеристики на независимите претенции и с характеристики на други зависими претенции, както е подходящо и не само както е изрично изложено в претенциите.

Горепосочените и други характеристики, особености и предимства на настоящия полезен модел ще станат очевидни от следното подробно описание, взето в съчетание с приложените чертежи, които илюстрират, чрез пример, принципите на полезния модел. Това описание е дадено само за пример, без да ограничава обхвата на полезния модел. Цитираните по-долу референтни фигури се отнасят до приложените чертежи.

#### Пояснение на приложените фигури

Фигура 1 е блокова схема на обобщен теоретичен модел на Big Data архитектура според предшестващото състояние на техниката.

Фигура 2 е блокова схема на Lambda архитектура за обработка на големи данни според предшестващото състояние на техниката.

Фигура 3 е блокова схема на Карра архитектура за обработка на големи данни според предшестващото състояние на техниката.

Фигура 4 е блокова схема на обобщен теоретичен модел на архитектура на Интернет-на-нещата според предшестващото състояние на техниката.

Фигура 5 е блокова схема на компютърно реализирана система за обработка на големи данни Big Data според предпочетен вариант на полезния модел.

Фигура 6 е схематично изображение на примерно изпълнение на набор от сензори и изпълнителни устройства, свързани към компютърно реализираната система за обработка на големи данни Big Data според предпочетен вариант на полезния модел.

Фигура 7 е блокова схема на хардуерно-софтуерен комплекс на компютърно реализирана система за обработка на големи данни Big Data според предпочетен вариант на полезния модел.

В различните фигури, същите референтни знаци се отнасят до същите или аналогични елементи.

#### **Списък с примерни елементи**

- 10 - първи/краен слой;
- 14 - IoT устройство;
- 16 - генеричен активатор - IDAS;
- 17 - компонент за обработка на контекстна информация според предварително създадени правила - PROTON;
- 18 - IoT агент;
- 19 - контекстен брокер за управление на контекстната информация - ORION;
- 20 - втори слой;
- 21 - Cygnus;
- 22 - обработващ подслой - COSMOS GE (генеричен активатор);
- 24 - съхраняващ подслой;
- 26 - Apache Spark за широкомащабна обработка на данни;
- 27 - конектор към Apache Spark за широкомащабна обработка на данни;
- 28 - Apache Flink за машинно обучение;
- 29 - конектор към Apache Flink за машинно обучение;
- 30 - трети слой/слой за приложения;
- 31 - База данни;
- 32 - компонент за трайно съхранение на данни от контекстния брокер - Draco GE;
- 34 - релационна база данни - MySQL;
- 35 - релационна база данни - PostgreSQL;
- 36 - NOSQL база данни - Mongo;

- 38 - компонент за публикуване, търсене и откриване на отворени набори от данни - SKAN;
- 42 - приложения;
- 44 - компонент за улесняване разработването на персонализирани оперативни табла за управление - WireCloud;
- 50 - високопроизводителен сървър;
- 51 - оперативна памет;
- 52 - многоядрен процесор - CPU Xeon Gold 5317;
- 54 - графичен процесор - NVIDIA L40;
- 55- Компоненти на FIWARE;
- 56 - дисков масив за съхранение на големи данни;
- 60 - Подсистема за връзка с външни устройства;
- 60.1 - Сензор за температура;
- 60.2 - Сензор за движение;
- 60.3 - Умна лампа;
- 60.4 - Умен термостат;
- 60.5 - Шлюз към крайни устройства;
- 62 - хардуерна част с входно/изходни контролери - WiFi контролер и USB контролер;
- 64 - софтуерна част с драйвери за прекодиране на данни между външните устройства и IoT агента - IDAS драйвери.

### **Примери за изпълнение на полезния модел**

Настоящият полезен модел ще бъде описано по отношение на конкретни изпълнения и с позоваване на определени чертежи, но полезният модел не е ограничено до тях, а само от претенциите. Описаните чертежи са само схематични и неограничаващи. В чертежите размерът на някои от елементите може да бъде преувеличен и не в мащаб и са само за илюстративни цели. Размерите и относителните размери не съответстват на действителни реализации на полезния модел.

Следните термини са предоставени единствено за подпомагане на разбирането на полезния модел.

Както е използвано тук и освен ако не е посочено друго, терминът "големи данни" ("Big Data") означава изключително големи и разнообразни обеми от структурирани и неструктурирани данни, които изискват усъвършенствани методи и технологии за събиране, съхранение, анализ и обработка. Примери за "големи данни" включват данни, генерирани от IoT устройства, социални медии, научни изследвания и други източници с висока скорост и обем.

Както е използвано тук и освен ако не е посочено друго, терминът "обработващ подслой" означава част от втория слой на системата, която е отговорна за обработката и анализа на данните, включително изпълнението на изчислителни задачи и алгоритми за машинно обучение. Примери за компоненти на "обработващ подслой" са Apache Spark за широкомащабна обработка на данни и Apache Flink за машинно обучение.

Както е използвано тук и освен ако не е посочено друго, терминът "съхраняващ подслой" означава част от втория слой на системата, която осигурява съхранението на данните чрез бази данни и други механизми

за трайно запазване. Примери за "съхраняващ подслој" включват релационни бази данни като MySQL и NoSQL бази данни като MongoDB.

Както е използвано тук и освен ако не е посочено друго, терминът "настройки" означава софтуерен механизъм или модул, конфигуриран да управлява и контролира работата на системата, включително превключване между различни режими на работа и управление на компонентите за обработка на данни. Реално настройките се намират в конфигурационни файлове с разширение .xml на дисковото пространство. Логически те се ползват от приложенията.

Както е използвано тук и освен ако не е посочено друго, терминът "трислоен режим" означава режим на работа на системата, при който първият, вторият и третият слой, включително обработващият и съхраняващият подслој, функционират независимо и свързано. В "трислоен режим" системата използва пълния си потенциал за обработка и съхранение на данни.

Както е използвано тук и освен ако не е посочено друго, терминът "двуслоен режим" означава режим на работа на системата, при който обработващият подслој и съхраняващият подслој са свързани или един от тях е деактивиран, в резултат на което системата функционира с два слоя. В "двуслоен режим" системата може да жертва част от функционалността за по-голяма бързина или ефективност.

Както е използвано тук и освен ако не е посочено друго, терминът "IoT устройство" означава електронно устройство, свързано към Интернет на нещата (IoT), което може да събира и предава данни през мрежа. Примери за "IoT устройство" включват сензори за температура, умни домашни уреди и индустриални контролери.

Както е използвано тук и освен ако не е посочено друго, терминът "HTTP адаптер" означава софтуерен компонент, който осигурява комуникация между IoT устройства и системата посредством HTTP протокола. "HTTP адаптер" позволява прехвърляне на данни чрез стандартни HTTP заявки.

Както е използвано тук и освен ако не е посочено друго, терминът "IoT агент" означава софтуерен компонент, който действа като посредник между HTTP адаптера и контекстния брокер, преобразувайки данни и протоколи. Пример за "IoT агент" е IDAS, който улеснява интеграцията на различни IoT устройства с контекстния брокер.

Както е използвано тук и освен ако не е посочено друго, терминът "контекстен брокер" означава софтуерен компонент, който управлява контекстната информация в системата, като осигурява централизирано съхранение и достъп до данните. Пример за "контекстен брокер" е Orion, който поддържа управление на контекстна информация чрез стандартни API.

Както е използвано тук и освен ако не е посочено друго, фразата "компонент за обработка на контекстна информация според предварително създадени правила" означава софтуерен модул, който анализира контекстни данни и изпълнява действия според дефинирани правила. Пример за такъв компонент е PROTON генеричен активатор, който обработва събития в реално време.

Както е използвано тук и освен ако не е посочено друго, терминът "PROTON генеричен активатор" означава софтуерен компонент за обработка на потоци от събития, позволяващ дефиниране на сложни събитийни правила и реакции. "PROTON генеричен активатор" се използва за откриване на модели и зависимости в данните.

Както е използвано тук и освен ако не е посочено друго, терминът "IDAS" означава IoT Data Agents Suite, която предоставя услуги за свързване на IoT устройства с контекстен брокер. "IDAS" улеснява интеграцията на различни протоколи и устройства.

Както е използвано тук и освен ако не е посочено друго, терминът "Orion" означава Orion Context Broker, софтуерен компонент за управление на контекстната информация чрез RESTful API. "Orion" позволява управление на създаване, актуализация и запитвания към контекстни данни.

Както е използвано тук и освен ако не е посочено друго, терминът "NGSILD обекти" означава модели на данни, съвместими със спецификацията NGSI-LD, която дефинира стандарт за управление на контекстна информация с използване на свързани данни (Linked Data). Използването на "NGSILD обекти" подобрява интероперативността и семантичното богатство на данните.

Както е използвано тук и освен ако не е посочено друго, терминът "генеричен активатор COSMOS GE" означава софтуерен компонент от платформата FIWARE, който предоставя услуги за обработка на големи обеми данни и анализ в реално време. "COSMOS GE" използва технологии като Apache Spark за изпълнение на разпределени изчисления.

Както е използвано тук и освен ако не е посочено друго, терминът "Apache Spark" означава рамка с отворен код за паралелна обработка на големи данни, осигуряваща бързи изчисления с използване на памет и клъстерно изчисление. "Apache Spark" се използва за трансформации на данни и машинно обучение.

Както е използвано тук и освен ако не е посочено друго, терминът "Apache Flink" означава рамка с отворен код за обработка на поточни и пакетни данни, предоставяща висока производителност и възможности за анализ в реално време. "Apache Flink" е подходяща за приложения с ниска латентност и висока паралелност.

Както е използвано тук и освен ако не е посочено друго, терминът "конектор" означава софтуерен компонент, който осигурява връзка и обмен на данни между две софтуерни системи или компоненти. "Конекторите" между Apache Spark и контекстния брокер позволяват интеграция на данни за обработка.

Както е използвано тук и освен ако не е посочено друго, фразата "компонент за трайно съхранение" означава софтуерен модул, който записва данни от контекстния брокер в устойчиво хранилище за последващо използване или анализ. Пример за такъв компонент е Draco генеричен активатор.

Както е използвано тук и освен ако не е посочено друго, терминът "релационни бази данни" означава системи за управление на бази данни, които използват релационен модел и позволяват съхранение и достъп до данни чрез SQL заявки. Примери за "релационни бази данни" са MySQL и PostgreSQL.

Както е използвано тук и освен ако не е посочено друго, терминът "NoSQL бази данни" означава системи за управление на бази данни, които не следват релационния модел и са проектирани за съхранение на големи обеми от неструктурирани или полуструктурирани данни. Пример за "NoSQL база данни" е MongoDB.

Както е използвано тук и освен ако не е посочено друго, фразата "компонент за публикуване, търсене и откриване на отворени набори от данни" означава софтуерен модул, който осигурява услуги за управление и достъп до публично достъпни данни, включително функции за качване, индексирание и търсене. Пример за такъв компонент е SKAN разширение.

Както е използвано тук и освен ако не е посочено друго, терминът "Draco генеричен активатор" означава софтуерен компонент от платформата FIWARE, който осигурява трайно съхранение на данни от контекстния брокер в различни бази данни. "Draco" поддържа запис в релационни и NoSQL бази данни.

Както е използвано тук и освен ако не е посочено друго, терминът "MySQL" означава релационна база данни с отворен код, използваща SQL за достъп и управление на данни. "MySQL" е широко използвана за уеб приложения и съхранение на структурирани данни.

Както е използвано тук и освен ако не е посочено друго, терминът "Mongo" означава MongoDB, NoSQL база данни с отворен код, която съхранява данни в BSON формат и осигурява гъвкавост и мащабируемост. "MongoDB" е подходяща за съхранение на големи обеми от неструктурирани данни.

Както е използвано тук и освен ако не е посочено друго, терминът "CKAN разширение" означава допълнителен модул към системата CKAN за управление на отворени данни, който добавя специфични функционалности или интеграции. "CKAN разширение" може да предоставя визуализации или допълнителни интерфейси за данни.

Както е използвано тук и освен ако не е посочено друго, терминът "Wirecloud генеричен активатор" означава платформа за създаване на уеб приложения и персонализирани табла за управление чрез комбиниране на различни виджети и оператори. "Wirecloud" улеснява разработката на интерактивни интерфейси.

Както е използвано тук и освен ако не е посочено друго, терминът "високопроизводителен сървър" означава компютърна система с висока изчислителна мощност, състояща се от множество процесори, графични процесори и голям капацитет за съхранение, проектирана за обработка на големи данни. Такъв сървър може да изпълнява сложни изчислителни задачи с висока скорост.

Както е използвано тук и освен ако не е посочено друго, фразата "подсистема за връзка с външни устройства" означава хардуерни и софтуерни компоненти, осигуряващи свързаност и комуникация между системата и външни устройства или сензори. Тази подсистема включва входно/изходни контролери и драйвери.

Както е използвано тук и освен ако не е посочено друго, фразата "хардуерна част с входно/изходни контролери" означава физически устройства и интерфейси, които позволяват свързване на външни устройства към системата. Тези контролери управляват входните и изходните сигнали.

Както е използвано тук и освен ако не е посочено друго, фразата "софтуерна част с драйвери за прекодиране на данни" означава софтуерни модули, които управляват хардуерните контролери и преобразуват данни между форматите на външните устройства и системата. Тези драйвери осигуряват съвместимост и комуникация.

Както е използвано тук и освен ако не е посочено друго, терминът "графични процесори" означава специализирани изчислителни единици (GPU), предназначени за паралелна обработка на данни, включително задачи извън графиката като машинно обучение. "Графичните процесори" ускоряват изчислителни процеси с висока паралелност.

Както е използвано тук и освен ако не е посочено друго, терминът "диск масив" означава група от дискови устройства, работещи заедно за осигуряване на голям капацитет за съхранение и подобрена

производителност чрез техники като RAID. "Дисковият масив" осигурява бърз достъп до големи обеми данни.

Както е използвано тук и освен ако не е посочено друго, терминът "многоядрени процесори" означава процесори с повече от едно изчислително ядро на един чип, позволяващи паралелна обработка на задачи. "Многоядрените процесори" подобряват производителността при многозадачност и сложни изчисления.

Както е използвано тук и освен ако не е посочено друго, фразата "софтуерният контрол е конфигуриран да превключва между трислойния и двуслойния режим в зависимост от изискванията на приложенията за бързина или точност на обработката" означава, че системата автоматично избира подходящия режим на работа според нуждите на приложенията, за да осигури оптимална производителност или прецизност. Например, за по-бърза обработка може да се използва двуслоен режим, докато за по-точни резултати се активира трислойният режим.

Полезният модел ще бъде разяснен с подробно описание на няколко варианта на изпълнение на полезния модел. Ясно е, че други варианти на изпълнение на полезния модел могат да бъдат конфигурирани според знанията на специалисти в областта, без да се отклоняват от техническото същност на полезния модел.

В първия аспект, настоящият полезен модел се отнася до компютърно реализирана система за обработка на големи данни BigData, включваща първи слой 10 за получаване на данни от източници на данни; втори слой 20 за обработка и съхранение на данни, включващ обработващ подслой 22 и съхраняващ подслой 24; и трети слой 30 за приложения, използващи обработените данни. Системата е конфигурируема чрез софтуерен контрол да работи в трислоен режим, при който обработващият подслой 22 и съхраняващият подслой 24 работят независимо и свързано, и в двуслоен режим, при който обработващият подслой 22 и съхраняващият подслой 24 са свързани или един от тях е деактивиран. Както е илюстрирано на фигура 5, системата представя блокова схема на такъв компютърно реализиран комплекс за обработка на големи данни според предпочетен вариант на полезния модел.

В едно предпочитано примерно изпълнение, първият слой 10 може да включва IoT устройство 14, генеричен активатор 16, контекстен брокер 19 за управление на контекстната информация, при което IoT устройството 14 е свързано посредством HTTP адаптер (IoT агент 18) за прехвърляне на данни от/към IoT устройството 14 и генеричния активатор 16 и контекстният брокер 19. Като контекстният брокер 19 е свързан с компонент 17 за обработка на контекстна информация според предварително създадени правила. Това позволява ефективно събиране и управление на данни от различни IoT източници. Както е показано на фигура 6, тя представя схематично изображение на примерна реализация на набор от сензори и изпълнителни устройства, свързани към системата. Показани са сензор за температура 60.1, сензор за движение 60.2, умна лампа 60.3, умен термостат 60.4 и шлюз към крайни устройства 60.5, което допринася за по-добро разбиране на функционалността на системата.

Компонентът 17 за обработка на контекстна информация може да бъде PROTON генеричен активатор, генеричният активатор 16 може да бъде IDAS, а контекстният брокер 19 за управление на контекстната информация може да бъде Orion. Допълнителните функции на компонента 17 - Proton са за обработка на контекстна информация според предварително създадени правила, което позволява автоматизирана обработка на данните въз основа на дефинирана бизнес логика. Използването на тези утвърдени компоненти

осигурява надеждност и съвместимост. Контекстният брокер 19 може да бъде конфигуриран да моделира контекстната информация като NGS-LD обекти, което позволява стандартизирано представяне на контекстните данни.

Обработващият подслоей 22 може да е генеричен активатор COSMOS GE и да включва Apache Spark 26 за широкомащабна обработка на данни; Apache Flink 28 за машинно обучение; конектори 27 и 29 за свързване на Apache Spark 26 и Apache Flink 28 с контекстния брокер 19. Използването на тези мощни инструменти за анализ на големи данни, осигурява висока производителност и гъвкавост.

В едно предпочитано примерно изпълнение, обработващите компоненти могат да използват графични процесори (GPU) за ускоряване на задачите по обработка на данни. В такива реализации, GPU могат да бъдат използвани от компоненти като Apache Spark 26 и Apache Flink 28 за ускоряване на пакетната и стрийминг обработка на големи набори от данни. В конкретни реализации, високопроизводителният сървър 50 може да съдържа поне два GPU 54, като например графични карти NVIDIA L40, и компонентите Apache Spark 26 и Apache Flink 28 в COSMOS GE могат да използват тези GPU 54 за ускоряване на изчисленията, свързани с анализа на данни в стрийминг и пакетен режим.

Съхраняващият подслон 24 може да включва компонент 32 за трайно съхранение на данни от контекстния брокер 19, свързан към релационна база данни 34, като MySQL и/или към NoSQL база данни 36 като MongoDB и към компонент 38 за публикуване, търсене и откриване на отворени набори от данни, като SKAN разширение. Това позволява надеждно съхранение и лесен достъп до данните, като използването на тези утвърдени технологии гарантира стабилност и съвместимост.

Третият слой 30 може да включва приложения 42, които комуникират директно с базите данни 34 и 36 или с компонентите на втория слой 20; компонент 44 за улесняване разработването на персонализирани оперативни табла за управление, като например Wirecloud генеричен активатор. Това осигурява мощен и гъвкав инструмент за визуализация и позволява лесно изграждане на потребителски приложения, използващи обработените големи данни.

В едно предпочитано примерно изпълнение, софтуерните компоненти могат да бъдат изпълнявани паралелно върху множество ядра на процесора за пълноценно използване на хардуерните ресурси и оптимална работа на системата. В такива реализации, системата може да включва високопроизводителен сървър 50 с множество многоядрени процесори 52, като генеричните активатори могат да бъдат разпределени върху ядрата на процесора от ядрото на операционната система за паралелно изпълнение, повишавайки производителността и ефективността. В конкретни реализации, високопроизводителният сървър 50 може да има поне два процесора 52, всеки с множество ядра (общо 32 ядра), като софтуерните компоненти, включително генеричните активатори, системата за управление на бази данни и IDAS драйверите, могат да бъдат заредени в RAM паметта на сървъра. Тези компоненти могат да бъдат изпълнявани от ядрата на процесора, както е определено от ядрото на операционната система, осигурявайки пълноценно използване на хардуера и оптимална съвместна работа на отделните активатори.

Както е илюстрирано на фигура 7, системата, може да включва високопроизводителен сървър 50 с поне два многоядрени процесора 52, поне два графични процесора 54 и дисков масив 56 за съхранение на големи данни. В едно предпочитано примерно изпълнение, високопроизводителният сървър 50 може да има конкретна хардуерна конфигурация за подобрена производителност при обработката на големи данни. В

такива реализации, сървърът 50 може да включва два процесора 52 Intel Xeon Gold 5317, всеки с 12 ядра и 24 нишки на 3.0GHz, 18 MB; 8 модула по 16GB DDR4/3200MHz RDIMM памет (общо 128GB RAM); две NVIDIA L40 GPU 54 с по 48GB VRAM, свързани чрез PCI-E 4.0 x16; две SSD диска 56 от 3.84TB SATA 6Gb за високоскоростно съхранение на данни. Това позволява високопроизводителна обработка и съхранение на големи данни чрез значителна изчислителна мощност и капацитет за съхранение.

Също така е показана подсистема за връзка с външни устройства 60 на фигури 6 и 7, състояща се от хардуерна част 62 с входно/изходни контролери, напр. WiFi контролер и USB контролер, и софтуерна част 64 с драйвери за прекодиране на данни между външните устройства (IDAS драйвери) и IoT агента 18. Използването на мощен хардуер осигурява висока производителност при обработката на големи обеми данни и позволява лесна интеграция с разнообразни външни системи и устройства.

Хардуерната част се състои от входно/изходни контролери като например USB, WiFi и други.

Софтуерната част се състои от набор от драйвери, които прекодират данни от устройствата, така че да бъдат разпознати от IDAS активатор (enabler) както и управляващите команди изпратени чрез IDAS активатор (enabler) в разпознаваеми такива от изпълнителните устройства, към които са отправени.

В едно предпочитано примерно изпълнение, софтуерните компоненти могат да бъдат конфигурирани да комуникират ефективно с хардуерните ресурси за оптимална обработка и съхранение на данните. В такива случаи, софтуерните компоненти като генеричните активатори, системата за управление на бази данни и IDAS драйверите могат да се намират в оперативната памет (RAM) на високопроизводителния сървър 50 и да бъдат изпълнявани от ядра на процесора 52, разпределени от ядрото на операционната система. Системата за управление на бази данни може да комуникира с дисковия масив 56 за четене и запис на големи данни, а компонентите на COSMOS, а именно Apache Spark 26 и Apache Flink 28, могат да използват GPU 54 за ускоряване на изчисленията при анализ на данни в стрийминг и пакетен режим. По този начин се осигурява ефективна комуникация между софтуерните компоненти и хардуерните ресурси за повишена производителност.

В едно предпочитано примерно изпълнение, системата може да бъде специално проектирана да адресира предизвикателства при обработката на разнообразни типове данни, интегрирането на различни източници на данни, осигуряването на сигурност и предоставянето на технологична независимост и отворена архитектура. В такива реализации, архитектурата може да бъде конфигурирана да обработва различни видове контекстна информация от умни градове или подобни среди; да работи с разнообразие от източници на данни, включително IoT устройства, потребителски входове и съществуващи системи; да поддържа сигурност в цялата архитектура при управлението на градски услуги или приложения на трети страни; да се интегрира гъвкаво със съществуващи или бъдещи системи без да влияе на тяхната архитектура; да остане технологично независима, позволявайки източниците на информация за даден обект да варират с течение на времето; да се свързва и комуникира лесно с външни устройства, включително умни устройства; да бъде от отворен тип, за да насърчава широка съвместимост и разширяемост.

В конкретни реализации, архитектурата може да реализира хибриден модел, комбиниращ характеристики на Lambda и Карпа архитектури, позволявайки превключване между трислоен и двуслоен режим в зависимост от изискванията на приложенията за бързина или точност. Системата може да използва отворен код, по-специално генерични активатори на FIWARE, за да осигури откритост и съвместимост. Тя

може да включва компоненти като Orion Context Broker 19, PROTON GE, COSMOS GE и други за ефективно управление и обработка на контекстна информация. Системата може да адресира редица конкретни технически предизвикателства, предоставяйки надеждно и гъвкаво решение за обработка на големи данни, адаптивно към различни приложения като управление на умни градове, контрол на трафика, обществена администрация и други области, които изискват работа с големи данни.

Във втория аспект, настоящият полезен модел се отнася до компютърно реализирана система за обработка на големи данни Big Data, при която софтуерният контрол е конфигуриран да превключва между трислойния и двуслойния режим в зависимост от изискванията на приложенията за бързина или точност на обработката. Превключването между трислоен и двуслоен режим може да осигури оптимален баланс между производителност и прецизност на обработката на големите данни, което позволява гъвкаво адаптиране към различни сценарии на използване.

Сега, като се позовем отново на фигура 5, можем да видим, че системата е конфигурируема чрез софтуерен контрол да работи в различни режими, осигурявайки гъвкавост и адаптивност в зависимост от конкретните нужди. Това е ключово за приложения, които изискват различни нива на бързина и точност.

Като обобщение, представената компютърно реализирана система за обработка на големи данни Big Data предоставя многослойна архитектура с възможност за конфигуриране и адаптиране към разнообразни изисквания. Чрез интегрирането на утвърдени хардуерни и софтуерни компоненти, системата осигурява надеждност, висока производителност и гъвкавост при обработката и управлението на големи обеми данни от различни източници.

**Пример 1:** Компютърно реализирана система за обработка на големи данни Big Data.

Предложена е компютърно реализирана система за обработка на големи данни Big Data, реализираща нова хибридна архитектура, съчетаваща предимствата на Lambda и Карпа архитектурите. Системата се състои от три основни слоя - външен слой за получаване на данни от различни източници, слой за обработка и съхранение на данните, разделен на два подслоя, и слой с приложения, използващи обработените данни.

Ключова характеристика на системата е възможността за софтуерно конфигуриране да работи в два режима - трислоен, при който обработващият и съхраняващият подслой работят независимо и свързано за по-висока точност, и двуслоен, при който двата подслоя са свързани или единият е деактивиран за по-висока скорост на обработка.

Външният слой включва IoT устройства, HTTP адаптер за прехвърляне на данни, IoT агент IDAS за комуникация с контекстния брокер Orion, който управлява цялата контекстна информация, и PROTON компонент за обработка на контекстна информация според предварително създадени правила.

Обработващият подслой се състои от Apache Spark за пакетна обработка (batch processing), Apache Flink за машинно обучение и конектори, свързващи ги с контекстния брокер. Съхраняващият подслой включва Dгасо компонент за трайно съхранение, релационни и NoSQL бази данни, и SKAN разширение за управление на отворени данни. Flink и Spark заедно или поотделно софтуерно реализират пакетната или скоростната обработка според настройка, съответстваща на целите на приложението.

Приложният слой (Application layer) съдържа приложения, комуникиращи директно с базите данни или обработващия подслой и Wirecloud компонент за създаване на персонализирани табла за управление.

Системата е реализирана на базата на високопроизводителен сървър с многоядрени и графични процесори и голям дисков масив. Включва и подсистема за връзка с външни устройства с хардуерни контролери и софтуерни драйвери за комуникация.

Софтуерният контрол превключва между трислойния и двуслойния режим според изискванията на приложенията за бързина или точност на обработката на големите данни. По този начин хибридната архитектура съчетава гъвкаво предимствата на утвърдените Lambda и Карра подходи.

Материали и методи: Системата използва следната хардуерна конфигурация - високопроизводителен сървър с два процесора Intel Xeon Gold с общо 32 ядра, 8 модула оперативна памет с общ капацитет 128 GB, два SSD диска с капацитет 7.68 TB, два графични процесора NVIDIA L40.

Софтуерната реализация се базира на компоненти с отворен код от екосистемата FIWARE. Използвани са генеричните активатори Orion, IDAS, PROTON, Draco, Wirecloud и др. Обработката и анализът на големи данни се извършва чрез Apache Spark и Apache Flink, интегрирани чрез конектори. Данните се съхраняват в MySQL и Mongo бази данни. Включена е подсистема за връзка с разнообразни IoT устройства чрез хардуерни контролери и софтуерни драйвери.

Конфигурирането на системата в трислоен или двуслоен режим се извършва чрез софтуерен контрол според изискванията на конкретните приложения за приоритет на бързината или точността на обработката.

Резултати: Предложената компютърно реализирана система имплементира нова хибридна архитектура за обработка на големи данни, съчетаваща по гъвкав начин предимствата на съществуващите Lambda и Карра подходи. Трислойната конфигурация с независими слоеве за обработка и съхранение осигурява по-висока точност, докато двуслойната конфигурация с обединени или редуцирани слоеве позволява по-висока скорост.

Използването на утвърдени софтуерни компоненти с отворен код от средата FIWARE и интеграцията на водещи решения като Apache Spark и Flink осигуряват ефективна обработка и анализ на големи данни. Системата поддържа разнообразни източници на данни, включително IoT устройства, и предоставя възможности за персонализирана визуализация на резултатите.

Хардуерната конфигурация с високопроизводителен сървър, многоядрени и графични процесори и голям дисков масив осигурява необходимите изчислителни ресурси и съхранение за работа с големи данни.

Анализ и дискусия: Предложената система адресира ключови предизвикателства при обработката на големи данни като съчетаването на бързина и точност, работата с разнообразни източници, интеграцията на утвърдени решения и персонализираното представяне на резултатите.

Хибридната архитектура предоставя гъвкавост чрез софтуерно превключване между режими с фокус върху точността или бързината според специфичните изисквания на приложенията. Използването на компоненти с отворен код и стандартни интерфейси улеснява разширяемостта и адаптивността на системата.

Високопроизводителната хардуерна конфигурация осигурява необходимата изчислителна мощ и съхранение, но може да бъде оптимизирана спрямо конкретните нужди. Подсистемата за връзка с IoT устройства позволява интеграция с разнообразна сензорика и изпълнителни механизми.

Бъдещото развитие на системата може да включва допълнителни механизми за управление на сигурността и поверителността на данните, авточувствителност към натоварването и автоматизирано мащабиране, както и разширени възможности за извличане на знания и подпомагане вземането на решения.

**Пример 2:** Едно възможно реално приложение на предлаганата компютърно реализирана система за обработка на големи данни Big Data е в областта на интелигентните градове.

Системата може да интегрира данни от множество разнородни източници в градската среда и да осигурява тяхната ефективна обработка, анализ и визуализация за нуждите на градското управление.

Външният слой на системата може да включва разнообразни IoT устройства, разположени в градската инфраструктура:

- Сензори за качество на въздуха, измерващи нива на замърсители като фини прахови частици, азотни оксиди, озон и др.;
- Сензори за шумово замърсяване в различни точки на града;
- Камери за видеонаблюдение на трафика и обществените пространства;
- Сензори за заетост на паркоместата;
- Интелигентни електромери и водомери;
- Сензори за нивото на отпадъците в контейнерите;
- Метеорологични станции.

HTTP адаптерът и IoT агентът IDAS ще осигуряват свързаността на тези разнородни устройства с контекстния брокер Orion. PROTON компонентът ще обработва постъпващата информация според предварително зададени правила, например за генериране на предупреждения при надвишаване на прагови стойности за замърсяване.

Обработващият подслой използва Apache Spark за анализ на големи обеми исторически данни с цел идентифициране на тенденции и зависимости. Apache Flink се прилага за обработка на потоци от данни в реално време, например за динамично управление на светофарите според текущата трафик ситуация.

Съхраняващият подслой използва Draco за трайно съхранение на данните в MySQL и MongoDB бази данни. SKAN разширението осигурява публичен достъп до подобрени набори от отворени данни за града.

Приложният слой включва редица специализирани приложения:

- Система за мониторинг на качеството на въздуха и генериране на предупреждения;
- Платформа за управление на градския трафик и паркирането;
- Система за оптимизация на събирането на отпадъци;
- Приложение за енергиен мениджмънт на обществените сгради;
- Интерактивна карта на града с визуализация на различни показатели.

Wirecloud компонентът позволява създаване на персонализирани табла за управление според нуждите на различните градски служби.

Системата работи в трислоен режим при извършване на комплексни анализи, изискващи висока точност, например при дългосрочно планиране на градското развитие. Двуслойният режим се използва за приложения, изискващи бърза реакция в реално време, като управлението на трафика.

Материали и методи: Използва се описаната в пример 1 хардуерна и софтуерна конфигурация. Допълнително са интегрирани специализирани библиотеки за анализ на пространствени данни и машинно обучение. Разработени са интерфейси за връзка със съществуващи градски информационни системи.

Очаквани резултати: Системата успешно може да интегрира данни от над 10 000 сензора и IoT устройства в градската среда, да обработва средно дневен обем от 5 ТВ данни. Могат да се реализират функции като управление на трафика, електронно правителство, сигурност в реално време на избрани територии, наблюдение и управление на водни ресурси, ранно известяване на опасни събития - пожари, аварии и др.

Анализ и дискусия: Предложената система демонстрира ефективността на хибридната архитектура при работа с разнородни източници на данни в сложна градска среда. Гъвкавото превключване между режими позволява едновременно поддържане на приложения за дългосрочен анализ и такива за реакция в реално време. Интеграцията на отворени данни насърчава прозрачността и иновациите в градското управление.

Бъдещото развитие може да включва по-широко прилагане на изкуствен интелект за предиктивна аналитика, както и разширяване на системата за поддръжка на междуградско сътрудничество и обмен на данни.

Следва да се разбере, че въпреки че предпочитаните изпълнения, специфичните конструкции и конфигурации, както и материалите, са обсъдени тук за устройства според настоящия полезен модел, могат да бъдат направени различни промени или модификации във формата и детайлите, без да се отклонява от обхвата на това изобретение. Например, всякакви формули, дадени по-горе, са само представителни за процедури, които могат да бъдат използвани. Функционалност може да бъде добавена или премахната от блоковите диаграми и операциите могат да бъдат разменени между функционалните блокове. Стъпки могат да бъдат добавени или премахнати към методите, описани в рамките на обхвата на настоящия полезен модел.

## Претенции

1. Компютърно реализирана система за обработка на големи данни Big Data, включваща:
  - първи слой (10) за получаване на данни от източници на данни;
  - втори слой (20), свързан с първия слой (10), за обработка и съхранение на данни, включващ обработващ подслой (22) и съхраняващ подслой (24), свързан с обработващия подслой (22);
  - трети слой (30), свързан с втория слой (20), за осигуряване на достъп на потребителски приложения, използващи обработените данни; характеризираща се с това, че компютърно реализираната система е конфигурируема чрез софтуерен контрол да работи в: - трислоен режим, при който обработващият подслой (22) и съхраняващият подслой (24) работят независимо и свързано; и
  - двуслоен режим, при който съхраняващият подслой (22) е условно изключен.
2. Компютърно реализираната система за обработка на големи данни, съгласно претенция 1, характеризираща се с това, че първият слой (10) включва:
  - IoT устройство (14);
  - IoT агент (18);
  - генеричен активатор (16);
  - контекстен брокер (19) за управление на контекстната информация;
  - компонент (17) за обработка на контекстна информация според предварително създадени правила, при което
    - IoT устройството (14) е свързано към IoT агента (18);
    - IoT агентът (18) е свързан към генеричния активатор (16);
    - генеричният активатор (16) е свързан към контекстния брокер (19) за управление на контекстната информация;
    - контекстният брокер (19) за управление на контекстната информация е свързан към компонента (17) за обработка на контекстна информация според предварително създадени правила.
3. Компютърно реализираната система за обработка на големи данни Big Data, съгласно претенция 2, характеризираща се с това, че компонентът (17) за обработка на контекстна информация е PROTON генеричен активатор, генеричният активатор (16) е IDAS, а контекстният брокер (19) за управление на контекстната информация е Orion.
4. Компютърно реализираната система за обработка на големи данни Big Data, съгласно претенция 3, характеризираща се с това, че контекстният брокер (19) е конфигуриран да моделира контекстната информация като NGSI-LD обекти.
5. Компютърно реализираната система за обработка на големи данни Big Data, съгласно която и да е от предходните претенции, характеризираща се с това, че обработващият подслой (22) е генеричен активатор COSMOS GE и включва:
  - Apache Spark (26) за широкомащабна обработка на данни, свързан с контекстния брокер (19) чрез конектор (27);
  - Apache Flink (28) за машинно обучение, свързан с контекстния брокер (19) чрез конектор (29).
6. Компютърно реализираната система за обработка на големи данни Big Data, съгласно която и да е от предходните претенции, характеризираща се с това, че съхраняващият подслой (24) включва:

- компонент (32) за трайно съхранение на данни от контекстния брокер (19), свързан към контекстния брокер (19) и към реляционна база данни (34);

- към и/или NOSQL база данни (36);

- и към компонент (38) за публикуване, търсене и откриване на отворени набори от данни.

7. Компютърно реализираната система за обработка на големи данни Big Data, съгласно претенция 6, характеризира се с това, че компонентът (32) за трайно съхранение е Draco генеричен активатор, реляционните бази данни (34) са MySQL, NOSQL базите данни (36) са Mongo, а компонент (38) за публикуване, търсене и откриване на отворени набори от данни е SKAN разширение.

8. Компютърно реализираната система за обработка на големи данни Big Data, съгласно която и да е от предходните претенции, характеризира се с това, че третият слой (30) включва:

- приложения (42), свързани директно с базите данни (34, 36) или с компонентите на обработващия подслон (20);

- компонент (44), свързан с приложенията (42), за улесняване разработването на персонализирани оперативни табла за управление.

9. Компютърно реализираната система за обработка на големи данни Big Data, съгласно претенция 8, характеризира се с това, че компонентът (44) за улесняване разработването на персонализирани оперативни табла е Wirecloud генеричен активатор.

10. Компютърно реализираната система за обработка на големи данни Big Data, съгласно която и да е от предходните претенции, характеризира се с това, че допълнително включва високопроизводителен сървър (50), свързан със слоевете (10), (20) и (30), с:

- поне два многоядрени процесора (52);

- поне два графични процесора (54);

- дисков масив (56) за съхранение на големи данни.

## Приложение: 7 фигури

### Литература

1. Orozova, D., I. Popchev (2020) Cyber-Physical-Social Systems for Big Data. Proceedings of the 21st International Symposium on Electrical Apparatus & Technologies (SIELA), Bourgas, Bulgaria, 2020. PP-1-4, doi: 10.1109/SIELA49118.2020.9167161

<https://ieeexplore.ieee.org/document/9167161>

2. Oussous, A., F. Benjelloun, A. A. Lahcen, S. Belfkih (2018) Big Data technologies: A survey. Journal of King Saud University-Computer and Information Sciences 30 (2018). pp. 431-448.

<http://dx.doi.org/10.1016/i.iksuci.2017.06.001>

3. Popchev, I., D. Orozova (2019) Towards Big Data Analytics in the E-learning Space Cybernetics and Information Technologies, vol. 19(3), 2019, pp. 16-25.

4. Chicalanov, A., Lyubenova, M., Nikolov, R. (2016) Future of Internet, Ecosystem Services and Sustainable Regional Development. Proceedings of the 4th Annual International Conference on Ecology, Ecosystems and

Climate Change, 11-14 July, 2016, Athens, Greece. Athens: ATINER'S Conference Paper Series, No: ECL2016-2135. URL Conference Papers Series: [www.atiner.gr/papers.htm](http://www.atiner.gr/papers.htm). ISSN: 2241-2891.

5. Popchev, I., I. Radeva (2019) Risk Analysis - an Instrument for Technology Selection. Engineering Sciences LVI. 2019, No. 4, pp. 5-20. ISSN:1312-5702 (Print), 2603-3542 (Online), DOI: 10.7546/EngSci.LVI.19.04.01.

6. Khan, M., M. Uddin, N. Gupta, (2014) Seven V's of Big Data understanding Big Data to extract value. Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education, 2014, pp. 1-5, doi: 10.1109/ASEEZone1.2014.6820689.

7. Offia, Ch, M. Crowe (2019) A theoretical exploration of data management and integration in organisation sectors. International Journal of Database Management Systems (IJDMS). vol. 11, No. 1, February 2019. pp. 37-56. DOI: 10.5121/ijdms.2019.11103

8. Tejada, Z. (2022) Big data architectures. Microsoft, <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/>

9. Kalipe, K., R. K. Behera (2019) Big Data Architectures: A Detailed and Application Oriented Analysis. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, vol. 8, No. 9, July 2019. pp. 2182-2190.

DOI:10.35940/ijitee.H7179.078919

10. Singh, K., R. K. Behera, J. K. Mantri (2019) Big Data Ecosystem: Review on Architectural Evolution. In: Abraham, A., Dutta, P., Mandal, J., Bhattacharya, A., Dutta, S. (eds) Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, vol 813. Springer, Singapore, [https://doi.org/10.1007/978-981-13-1498-8\\_30](https://doi.org/10.1007/978-981-13-1498-8_30)

11. Marz, N., J. Warren (2015). Big Data: Principles and best practices of scalable real-time data systems. Manning, 2015. 330 p. ISBN: 9781617290343.

12. Kreps, J. (2014) Questioning the Lambda Architecture. O'Reilly.com, 2 July 2014; [www.oreilly.com/ideas/questioning-the-lambda-architecture](http://www.oreilly.com/ideas/questioning-the-lambda-architecture)

13. Sanla, A., T. Numnonda (2019) A Comparative Performance of Real-time Big Data Analytic Architectures. 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC). July 2019, pp. 1-5

doi: 10.1109/ICEIEC.2019.8784580

14. Hausenbias, M. (2015) A modern IoT data processing toolbox. 24 Apr. 2015. [https://fr.slideshare.net/Hadoop\\_Summit/a-modern-iot-data-processing-toolbox](https://fr.slideshare.net/Hadoop_Summit/a-modern-iot-data-processing-toolbox)

15. Alshuqayran, N., N. Ali, R. Evans (2016) A Systematic Mapping Study in Microservice Architecture. 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), 2016, pp. 44-51, doi: 10.1109/SOCA.2016.15

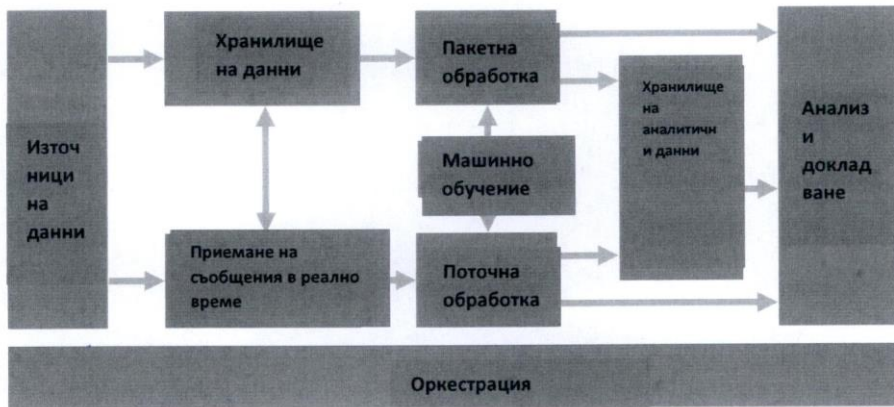
16. Scott, J. (2015) Zeta Architecture - What's in the Name? <https://medium.com/the-ramp/zeta-architecture-what-s-in-the-name-b2f68b42b94a>

17. FIWARE foundation, <https://digital-strategy.ec.europa.eu/en/news/fiware-foundation-research-those-digital-services-you-will-love>

18. Alonso A., A. Pozo, J. Cantera, F. De la Vega, J. Hierro (2018) Industrial Data Space Architecture Implementation Using FIWARE. Sensors (MDPI Basel). 2018 July 11. Vol. 18(7). <https://doi.org/10.3390/s18072226>.

19. Salhofer, P. (2018) Evaluating the FIWARE Platform: A Case-Study on Implementing Smart Application with FIWARE. Proceedings of the 51st Hawaii International Conference on System Sciences. URI: <http://hdl.handle.net/10125/50615>

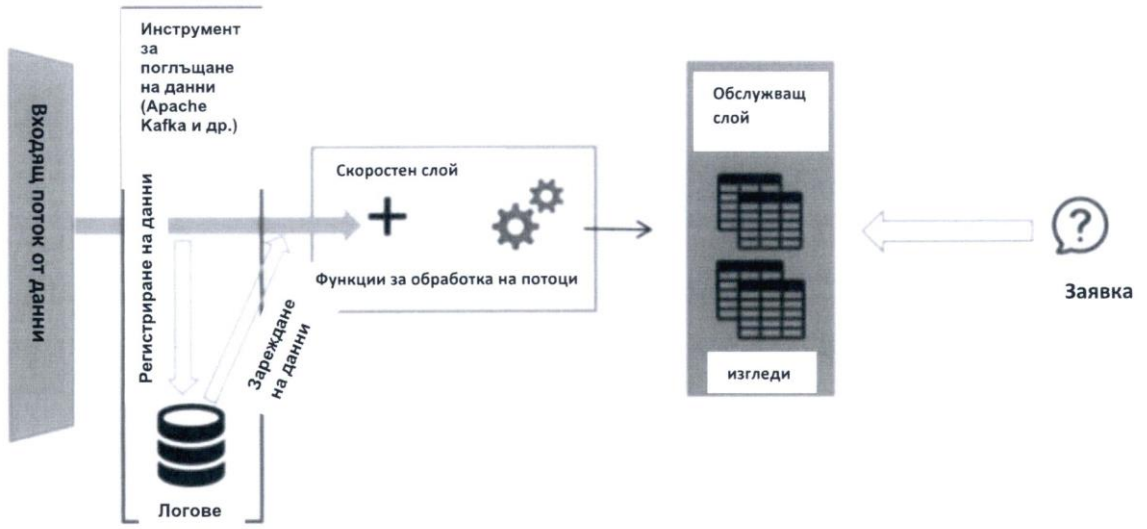
doi: 10.24251/HICSS.2018.726



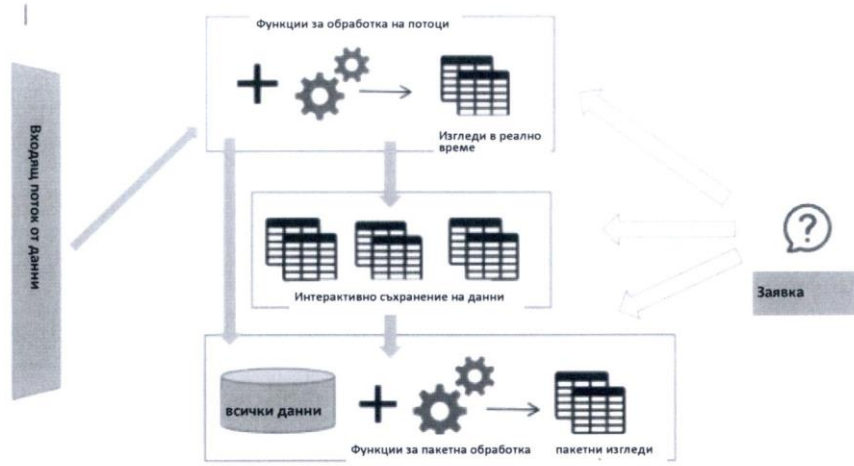
Фиг. 1



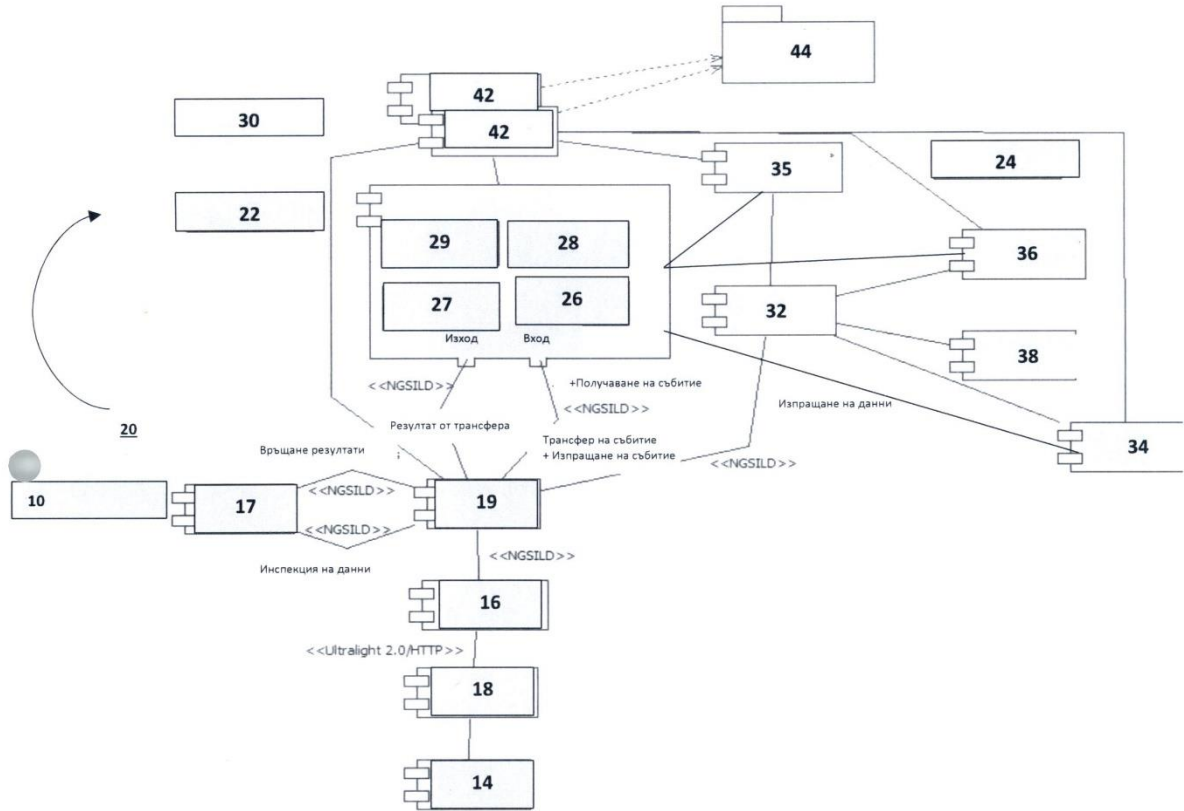
Фиг. 2



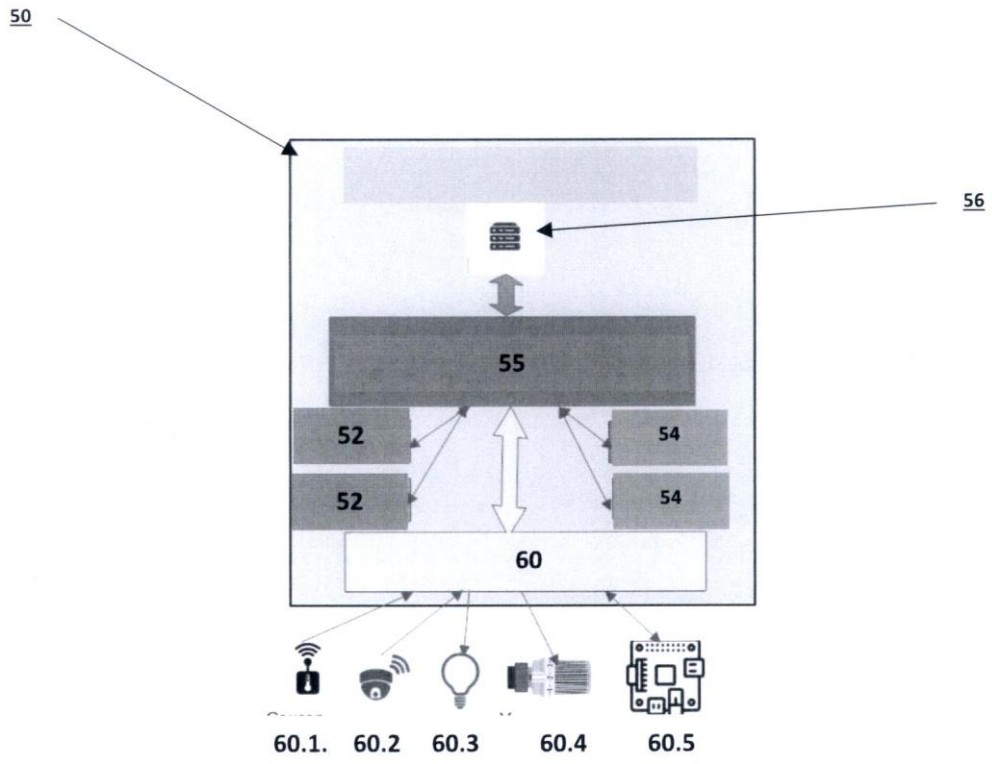
Фиг. 3



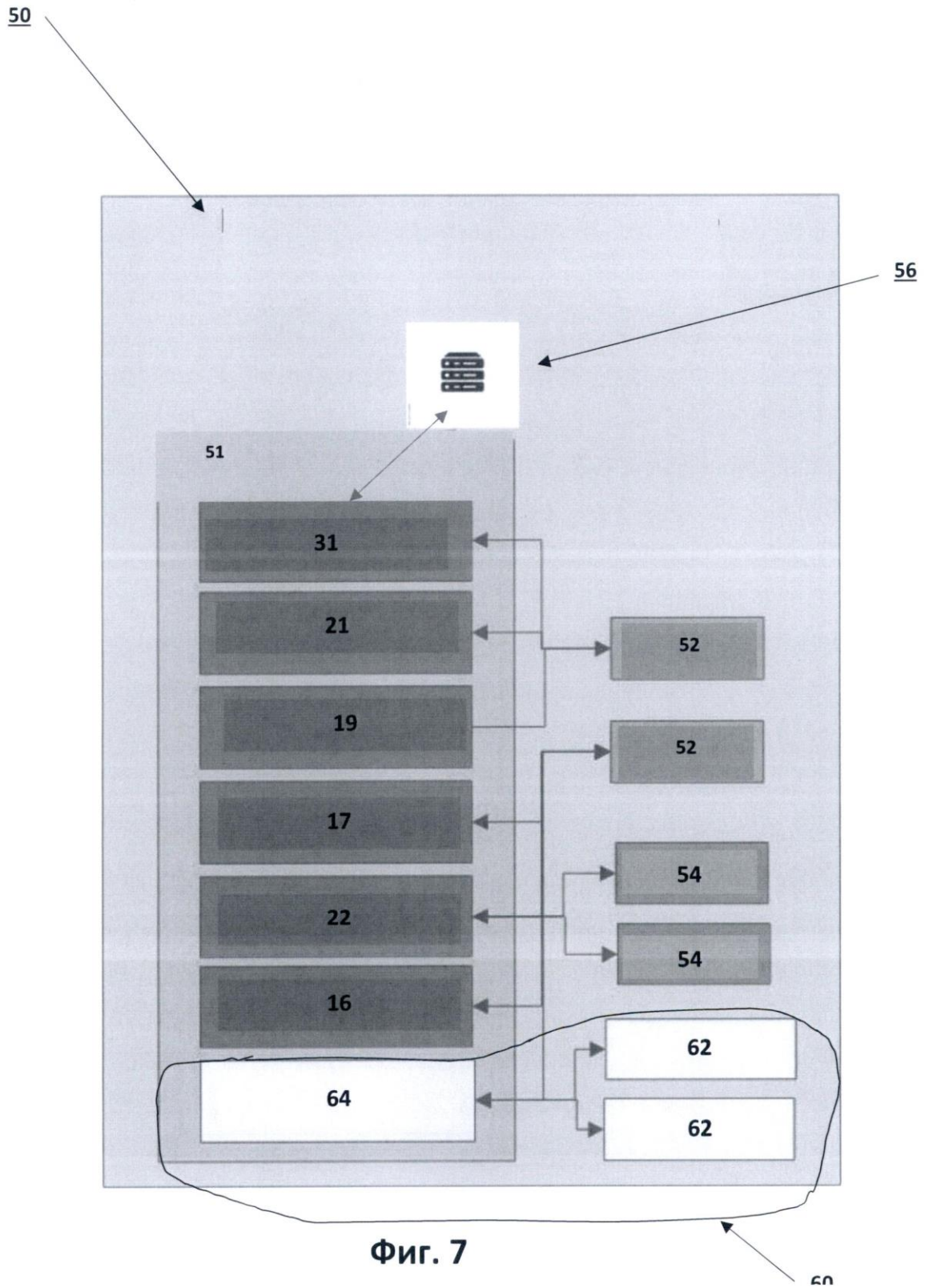
Фиг. 4.



Фиг. 5



Фиг. 6



Фиг. 7