

можуть конвертуватися в інші поширені формати (природно, з урахуванням особливостей обох форматів). З іншого боку, можливе його ефективно застосування при прийомі даних.

#### Використані джерела

УКРМАРК:

<http://www.nbuv.gov.ua/library/ukrmark.html>

Блок 6:

<http://www.nbuv.gov.ua/books/2002/ukrmarc/block6.html>

FRBR:

<http://www.ifla.org/VII/s13/frbr/frbr.pdf>

MARC21:

<http://www.loc.gov/marc/>

Classification and Call Number Fields (05X-08X):

<http://www.loc.gov/marc/bibliographic/lcbdclas.htm>

Subject Access Fields (6XX):

<http://www.loc.gov/marc/bibliographic/lcbdsbj.htm>

Added Entry Fields (70X-75X)

<http://www.loc.gov/marc/bibliographic/lcbdadde.htm>

MARC21 Format for Classification Data (2001 Concise Edition):

<http://www.loc.gov/marc/classification>

MARC Code Lists for Relators, Sources, Description Conventions:

Part III. Classification Sources:

<http://www.loc.gov/marc/relators/relaclas.html>

Part IV. Term, Name, Title Sources:

<http://www.loc.gov/marc/relators/relasour.html>

UKMARC:

<http://www.bl.uk/serveces/bibliographic/marc/>

UNIMARC:

Block 6 (Subject Analysis Block):

<http://www.ifla.org/VI/3/p1996-1/uni6.htm>

Appendix G (Subject System Codes)

<http://www.ifla.org/VI/3/p1996-1/appx-g.htm>

UNIMARC concise bibliographic format (1 Mar 2002):

<http://www.ifla.org/VI/3/p1996-1/concise2.pdf>

Concise UNIMARC Classification Format (20001031):

<http://www.ifla.org/VI/3/p1996-1/concise.htm>

Переклад українською мовою:

<http://www.nbuv.gov.ua/books/2002/ukrmarc/ukrclas.html>

UNIMARC Manual Authority Format 2001 (Concise Version)

<http://www.ifla.org/VI/p2001/guideright.htm>

UNIMARC Holdings Format (20 Mar 2003) (Concise Version)

<http://www.ifla.org/VI/3/nd1/unimarc-holdings.htm>

БЕЛМАРК

Блок 6:

<http://opac.bas-net.by/marc-doc/belmarc99/block6.html>

РУСМАРК:

Блок 6:

<http://www.rba.ru:8101/rusmarc/rusmarc/format1.htm>

Российский формат представления классификационных данных (Российская версия UNIMARC Classification Format):

<http://www.rba.ru:8101/rusmarc/commun>

Dublin Core:

<http://dublincore.org/documents/dces>

ONIX

<http://www.editeur.org/onixfles2.0/ONIXProductRecord2.0.pdf>

#### Список літератури

1. *Багрий І.* «Нам потрібна книга про...» або тематичний пошук в ЕК НБУВ // *Бібл. вісник.* – 1997. – № 5 – С. 11–13.
2. *Костенко Л. И.* Компьютерные технологии в Национальной библиотеке Украины имени В. И. Вернадского // *Библиотека и доступность информации в современном мире: электронные ресурсы науке, культуре и образованию.* Мат-лы междунар. конф. «Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества». Судак, 7–15 июня 2003 г. – Т. 1. – М.: Изд-во ГПНТБ России, 2003. – С. 217–219.

Денис СОЛОВЯНЕНКО

## Концептуальна модель системної архітектури онлайнного бібліотечного сервісу

Розглянуто питання працездатності системи ОБС. Визначено базові поняття ОБС: сервер ОБС (СОБС) та систему ОБС, їх призначення, умови функціонування та системну архітектуру. Побудовано концептуальну модель системної архітектури ОБС. Виділено та проаналізовано структурні елементи цієї моделі (рівні та підсистеми). Подано їхній детальний опис. Охарактеризовано механізми функціонування основних структурних компонентів СОБС та системи ОБС.

Останнім часом вітчизняні та, особливо, зарубіжні бібліотеки почали приділяти багато уваги наданню бібліотечних послуг в онлайнному режимі. Сьогодні на зміну невеличким бібліотечним сайтам приходять потужні інформаційні портали, здатні забезпечувати всі інформаційні запити користувачів із максимальною точністю та зручністю. Мільйони людей в усьому світі вже змогли оцінити зручність та надійність таких проектів.

Соловяненко Денис Володимирович, аспірант НБУВ.

Проте теоретично це питання ще мало досліджене. Технічною розробкою таких проектів в основному займаються не фахівці бібліотечної справи. Тому ці проекти часто створюються без урахування бібліотечної специфіки.

Сьогодні ситуація дещо змінюється. До наукового обігу вводиться поняття онлайнного бібліотечного сервісу (ОБС) як суттєво нового виду інформаційного сервісу, у якому поєднані традиційні бібліотечні та сучасні електронні технології. Проте цьому питанню присвяче-

но мало спеціальної літератури, особливо не вистачає фундаментальних бібліотечно-научних праць.

У цій роботі зроблено спробу побудувати концептуальну модель системної архітектури ОБС, визначити суб'єкти цього виду сервісу та розглянути їхню взаємодію.

ОБС, як і будь-який інший вид сервісу, що надається засобами комп'ютерної мережі, характеризується достатньо складною структурою. Це пов'язано з багатьма чинниками. Зручність, надійність, безпека та захист інформації, ергономічність – ось лише декілька з них. До того ж, у процесі реалізації ОБС беруть участь як мінімум чотири комп'ютери (три бібліотечні та один комп'ютер користувача). Це «теоретичний мінімум». На практиці ж комп'ютерів, які одночасно взаємодіють між собою в процесі ОБС, буває на чотири-п'ять порядків більше. Тому, для забезпечення безвідмовності цього процесу, структура ОБС має бути якомога чіткішою та прозорішою.

Але, перш ніж говорити про модель системної архітектури ОБС, потрібно визначити такі базові поняття, як сервер ОБС (СОБС) та система ОБС.

### СОБС: базові поняття

Дуже важливе розуміння поняття СОБС, адже саме СОБС є тією точкою доступу, за допомогою якої кінцевий користувач отримує потрібну йому інформацію з електронного фонду бібліотеки. Сьогодні поняття «сервер» використовується у декількох значеннях: 1) «комп'ютер, який надає свої ресурси, наприклад, принтери, файли, іншим комп'ютерам у мережі»<sup>1</sup>; 2) «програма на сітьовому комп'ютері, що відповідає за запити від клієнтських програм, які звертаються до серверу з інших сітьових комп'ютерів»<sup>2</sup>. Коли ми сервером називаємо комп'ютер (або комп'ютерну систему), ми говоримо про **фізичний сервер**. Коли ж ми називаємо сервером програму (або сукупність програм), маємо на увазі **логічний сервер**. СОБС є сервером в обох значеннях цього терміна.

Як фізичний сервер, **СОБС** являє собою комп'ютерну систему (один або декілька комп'ютерів з *адміністративного домену бібліотеки*), яка надає **ОБС** іншому комп'ютеру або системі комп'ютерів, що виступають як **клієнт ОБС (КОБС)**. Тут під **адміністративним доменом** розуміємо всю сукупність комп'ютерних систем та мереж, які знаходяться під адміністративним керівництвом бібліотеки. Та частина *адміністративного домену*, яку бібліотека виділила для надання ОБС, називається **доменною зоною СОБС**. Слід зазначити, що *доменна зона СОБС* може складатись лише з одного комп'ютера або включати в себе як одну, так і декілька локальних обчислювальних мереж з *адміністративного домену бібліотеки*. Єдиною вимогою до мережі є те, що вона повинна бути серверною, тобто один або де-

<sup>1</sup> Мирончиков И. К., Павловцев В. А. Англо-русский толковый словарь по сети «Интернет». – Минск: САДИ, 1997. – С. 76.

<sup>2</sup> Там само.

кілька вузлів повинні мати статус виділених серверів для обслуговування та маршрутизації запитів. Ці сервери приймають на себе всі функції зі спілкування з **КОБС**. Цей факт є важливим з огляду на те, що **КОБС** може звертатись до будь-якого комп'ютера з *доменної зони СОБС*, як до єдиного логічного **СОБС**.

Як логічний сервер **СОБС** являє собою сукупність програмного забезпечення для надання у режимі реального часу ОБС віддаленим користувачам комп'ютерної мережі. Логічний **СОБС** включає в себе багато програмних компонентів, проте головним серед них є система **ОБС**.

Слід зазначити, що один фізичний сервер може включати в себе багато логічних серверів, проте й один логічний сервер може об'єднувати в собі багато фізичних серверів. Це цілком виправдано для **СОБС**. На фізичному (апаратному) рівні до доменної зони **СОБС** може входити велика кількість фізичних серверів, проте **КОБС** може мати до них доступ, як до єдиного логічного **СОБС**. Можлива й зворотна ситуація. Бібліотека може розмістити на одному фізичному сервері велику кількість логічних **СОБС**, доступ до яких буде відбуватися по-різному<sup>3</sup>.

### Система ОБС: базові поняття

Оскільки **ОБС** реалізується без людського втручання, функції бібліотекаря щодо реалізації бібліотечного сервісу бере на себе програмна система – система **ОБС**. Ця система виконує всі ті ж функції, що і бібліотекар-людина (приймає користувачів, виконує їхні запити, звертається до бібліотечного фонду, шукає потрібні користувачеві ресурси та фіксує дані щодо наданих послуг і відмовлень). Проте, не слід ототожнювати систему **ОБС** з усім адміністративним доменом бібліотеки. Система **ОБС** – це «електронний бібліотекар», який є достатньо відокремленим від електронного фонду бібліотеки та тих апаратних та програмних засобів бібліотеки, які напряму не пов'язані з наданням **ОБС**. Система **ОБС** виконує функції посередника між електронним фондом бібліотеки та кінцевим користувачем інформації. Не слід ототожнювати систему **ОБС** із **СОБС**. Система **ОБС** є складовою частиною **СОБС**, що на апаратному рівні забезпечує роботу системи **ОБС**, а на програмному рівні крім цієї системи включає в себе також програмні засоби для адміністрування роботи системи **ОБС** та деякі додаткові елементи.

**Система ОБС** – це організована сукупність програмних засобів **СОБС**, що є точкою доступу віддаленого користувача комп'ютерної мережі для отримання **ОБС**. Система **ОБС** є основною частиною **СОБС** як логічного сервера.

**СОБС** надає системі **ОБС** апаратні, комунікаційні та програмні можливості для реалізації **ОБС**. Тобто призначенням **СОБС** є забезпечення функціонування системи **ОБС** – хостинг (hosting) системи **ОБС**.

<sup>3</sup> Детальніше про це див. у частині «Характеристика рівнів моделі системної архітектури **ОБС**».

### Модель системної архітектури ОБС

Принципово важливим кроком створення будь-якої обчислювальної системи є моделювання її системної архітектури. Для отримання базових понять майбутньої системи важливо побудувати її концептуальну модель. Концептуальна модель – це словесний опис природи та параметрів елементарних явищ досліджуваної системи, видів та засобів взаємодії між ними, значень кожного елементарного явища у загальному процесі функціонування цієї системи.

Перед тим, як втілювати в життя проект системи ОБС, потрібно розробити концептуальну модель її системної архітектури. Для цього системну архітектуру ОБС потрібно структурувати в межах сукупності окремих компонентів, тобто забезпечити потрібний рівень деталізації моделі. Це можна зробити шляхом побудови ієрархічної послідовності моделей, кожна з яких має відображати взаємодію між компонентами системної архітектури ОБС на різних рівнях деталізації.

Отже, маємо вертикальний поділ моделі на рівні. Горизонтальний поділ моделі – це поділ на підсистеми, кожна з яких відповідає за надання тієї чи іншої сукупності однорідних бібліотечних сервісів в онлайнному режимі.

Звісно, неможливо змоделювати кожен конкретний випадок надання ОБС, проте, якщо проаналізувати взаємодію, яка відбувається між основними структурними компонентами системної архітектури ОБС, можна побудувати типову модель ОБС<sup>4</sup>, що відобразить найбільш загальні властивості ОБС-взаємодії.

Отже, схематично концептуальну модель системної архітектури ОБС можна представити у вигляді пирога, де «кусками» будуть підсистеми, а шарами – рівні ОБС (схема 1).

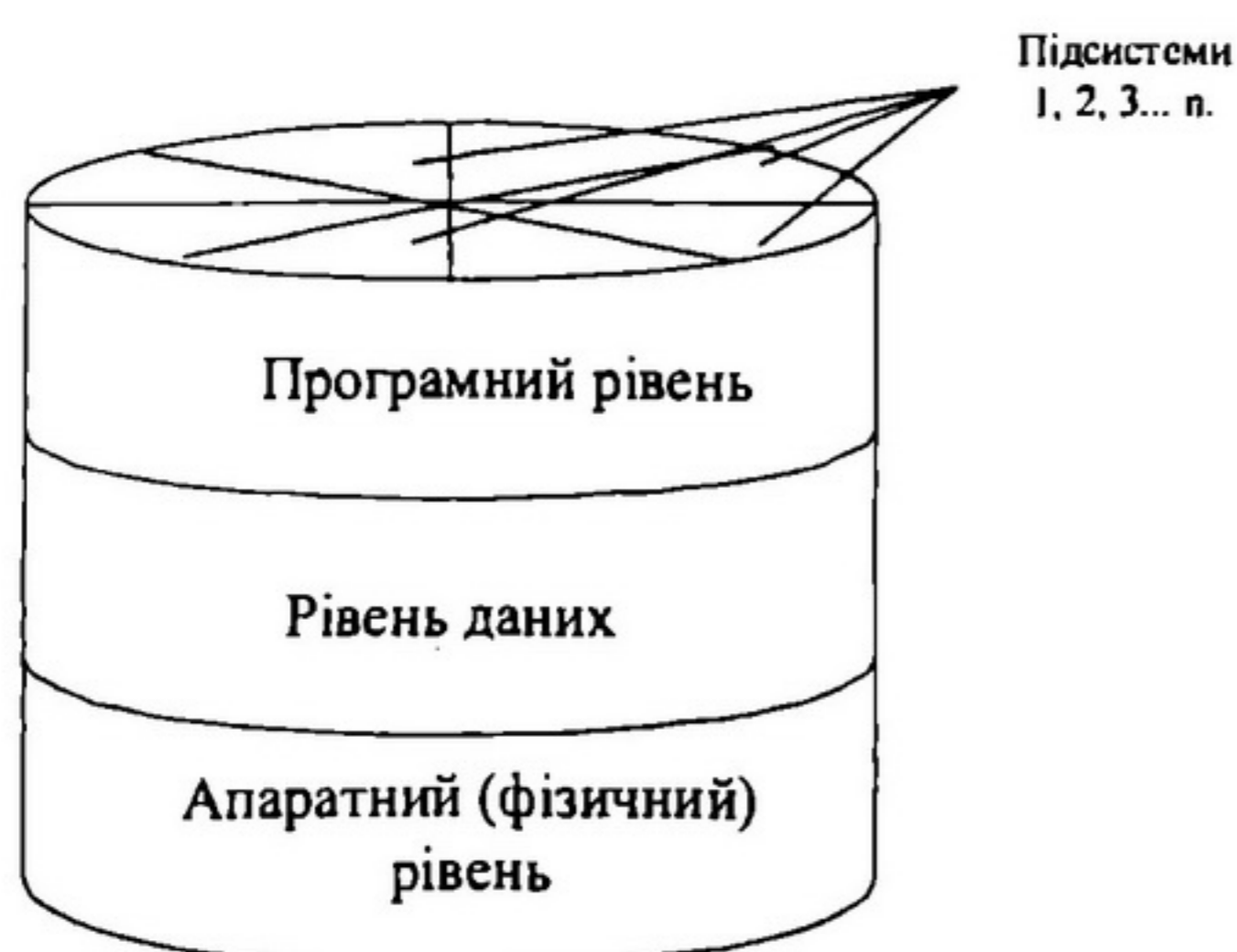


Схема 1. Концептуальна модель ОБС

ОБС реалізується на трьох рівнях, кількість же підсистем може бути різною (чотири і більше) і залежить від складності системи ОБС. Рівні моделі ОБС мають загальносистемний характер, тобто кожна з існуючих підсистем обов'язково реалізується на кожному з трьох рівнів.

<sup>4</sup> Слід зазначити, що тут мова йде не про еталонну модель OSI, яка є вірною для всіх відкритих систем, а про іншу модель, яка характеризує відмінність ОБС від інших видів сітьової взаємодії.

Розглянемо кожен із рівнів моделі ОБС.

### Характеристика рівнів моделі системної архітектури ОБС

Кожен із рівнів моделі ОБС відображає свою структуру реалізації ОБС. Тобто, на кожному рівні моделі схема процесу реалізації ОБС буде різною.

**Апаратний (фізичний) рівень** є базовим і відображає ту взаємодію, яка відбувається між фізичними об'єктами (комп'ютерами, фізичними серверами та іншими додатковими пристроями). Тобто компонентами системної архітектури на цьому рівні є фізичні пристрої. Схема апаратного рівня (схема 2) відображає фізичне розташування комп'ютерної техніки, що бере безпосередню або опосередковану участь у процесі реалізації ОБС.

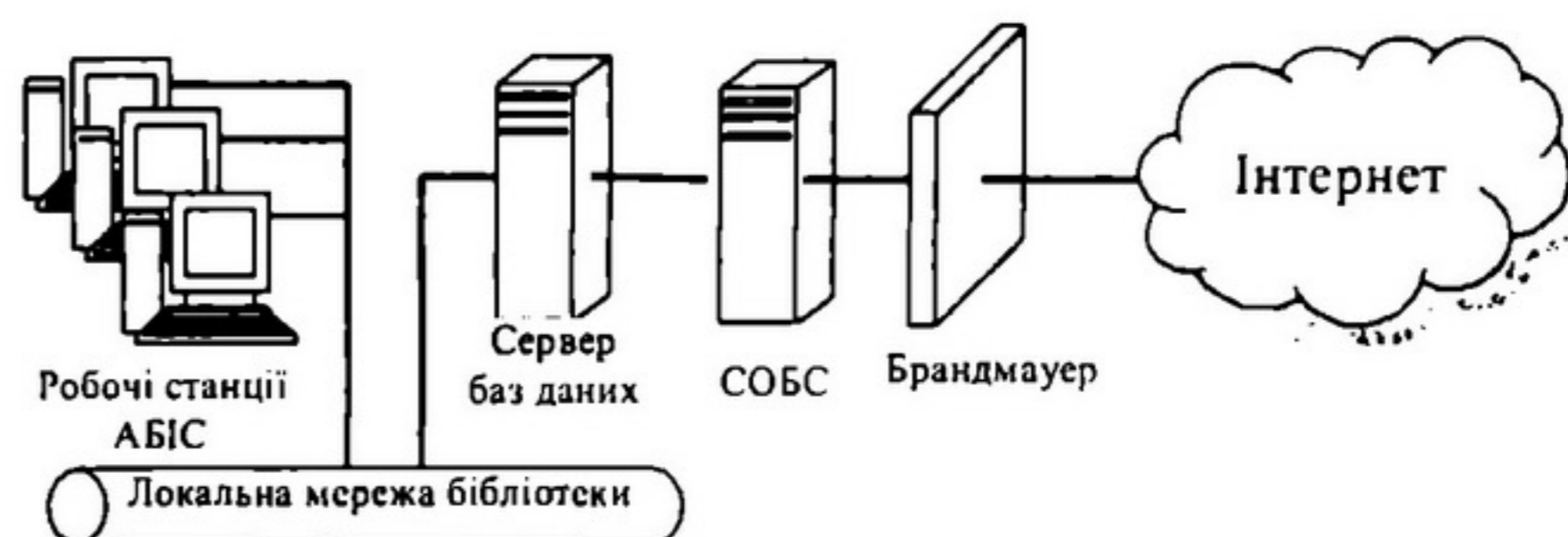


Схема 2. Апаратний (фізичний) рівень моделі системної архітектури ОБС

Як бачимо, для реалізації ОБС потрібна досить значна кількість комп'ютерної техніки. Основними елементами є виділені сервери. Як мінімум, повинно бути два фізичні сервери:

- ▼ Сервер баз даних (СБД) – умовно цей сервер можна назвати «електронним фондом»;
- ▼ СОБС, на якому розташована система ОБС, шлюз HTTP-Z 39.50 та так звані «рідні» БД – умовно цей сервер можна назвати «електронним бібліотекарем».

На схемі також присутній брандмауер. Слід зауважити, що цей елемент не є обов'язковим на даному рівні, проте він є бажаним. Брандмауер – це «засіб захисту серверів та мереж, що працюють в Інтернеті від несанкціонованого доступу. Запобігає попаданню в об'єкт, що захищається, або виходу з нього пакетів даних, які не відповідають встановленим правилам безпеки»<sup>5</sup>. Брандмауери бувають апаратні та програмні, надійніший захист забезпечує апаратний брандмауер.

Проте, апаратний брандмауер коштує десятки тисяч доларів, такі витрати може собі дозволити не кожна бібліотека. Програмний брандмауер (firewall) коштує набагато дешевше, але і надійність його набагато менша. Тому кожна бібліотека повинна сама вирішувати, який брандмауер їй потрібен: апаратний чи програмний. Слід врахувати й те, що безпека всієї мережі залежить від найслабкішого місця у ній. Тому, якщо бібліотека вже придбала апаратний брандмауер, то потрібно забезпечити й те, щоб усі зовнішні канали були захищені ним. В іншому разі майже всі переваги його використання

<sup>5</sup> Воройский Ф. С. Информатика. Новый систематизированный толковый словарь-справочник: Вводный курс по информатике и вычислительной технике в терминах. – 2-е изд., перераб. и доп. – М.: Либерия, 2001. – 536 с.

втрачаються. Якщо, наприклад, робочі станції АБІС мають прямий не захищений брандмауером вихід в Інтернет, то порушення безпеки СБД (або навіть СОБС) може відбутись (і, наймовірніше, відбудеться) саме з цього каналу.

Отже, присутність брандмауера на схемі цього рівня є вірною лише у тому разі, коли бібліотека використовує апаратний брандмауер.

Ще одним дуже важливим вузлом у наведеній схемі є автоматизована бібліотечно-інформаційна система (АБІС).

Як видно зі схеми, АБІС не має безпосереднього доступу до СБД, а із СОБС вона взагалі не може взаємодіяти. Цей факт є дуже важливим і пояснюється одразу трьома факторами. По-перше, цього вимагає політика безпеки СБД. На СБД зберігаються всі найважливіші дані (електронний фонд бібліотеки), втрата яких загрожує фатальними наслідками. Тому політика безпеки СБД повинна бути виключною, тобто доступ ззовні повинен реалізуватись лише у архітектурі «клієнт/сервер», причому жодний зовнішній клієнт (комп'ютерна система або програмний процес) не повинен мати виключних прав доступу. Виключні права доступу до СБД, які потрібні для адміністрування його роботи, повинні надаватись вкрай обмежено. Тобто, такий доступ має реалізовуватись лише при безпосередньому керуванні комп'ютером. Отже, СБД повинен вміти розпізнавати «рідні» та зовнішні програмні процеси та, відповідно, контролювати власну безпеку. Звісно, порушення безпеки передбачається, в основному, і з боку КОБС (тому для програмних процесів з Інтернету встановлено два рівні захисту: брандмауер, який захищає сервери від атак, та програмні засоби СОБС, які контролюють відкриття, проходження та закриття сеансів зв'язку), але АБІС, яка працює з БНД та БД СБД, теж прямо або побічно може становити загрозу безпеці даних на сервері, тому доступ обмежується з двох боків. Програмні процеси з робочих станцій АБІС мають більші привілеї у порівнянні з програмними процесами із СОБС, але виключних прав не отримують ні перші, ні другі.

Другим фактором, що зумовлює необхідність відокремлення АБІС від СБД та СОБС, є те, що сама АБІС потребує захисту від користувачів, і оскільки користувачі не мають прямого доступу до СБД (тобто запити користувачів маршрутизуються на СОБС, який не має ніяких прав доступу до АБІС), відповідно, АБІС стає повністю захищеною від них. І третій фактор. Якщо включити СБД до фізичної структури АБІС, із можливостями прямого доступу до нього, тобто якщо не робити його виділеним сервером, це суттєво знизить сільовий трафік та погіршить його роботу.

Отже, ми розглянули апаратний (фізичний) рівень моделі системної архітектури ОБС. Тепер розглянемо, як цей сервіс представлено на двох вищих рівнях.

**Рівень даних** відображає логічну схему доступу до даних із різних фізичних структур. Схема 3 відображає розташування БНД та БД на серверах бібліотеки, процес адміністрування цих БНД та БД (наповнення, кори-

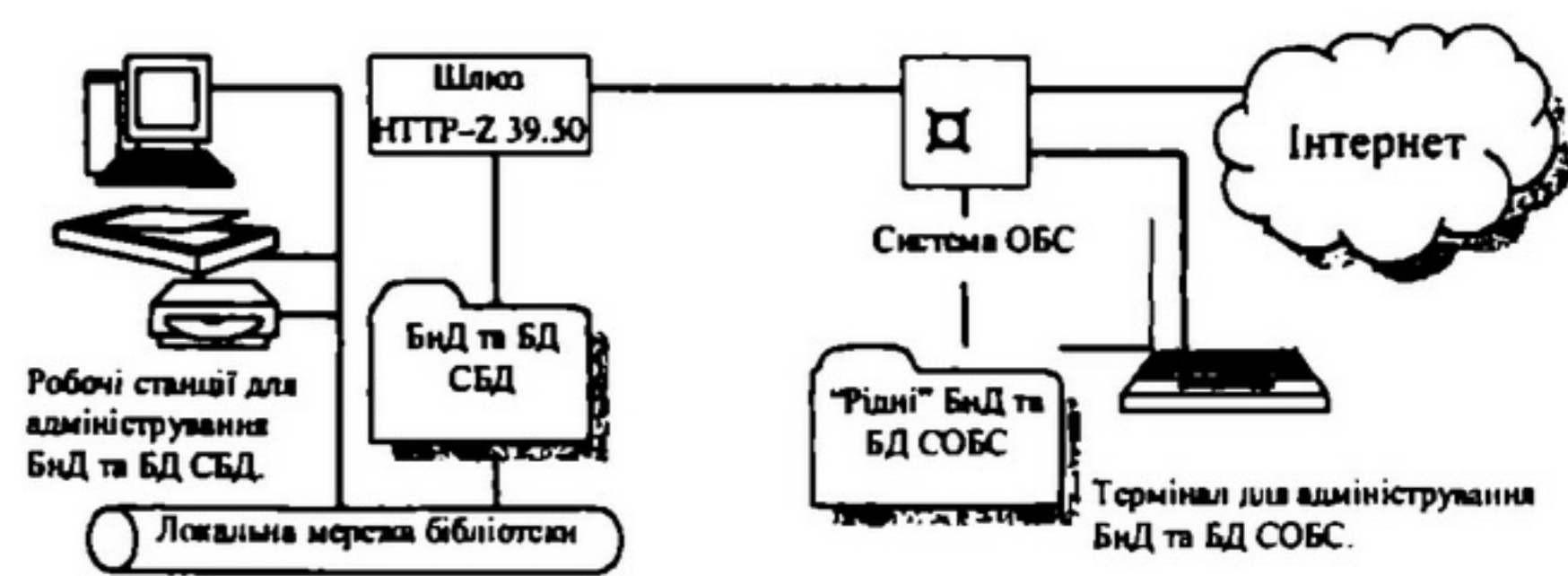


Схема 3. Рівень даних моделі системної архітектури ОБС

гування, видалення) та процес доступу КОБС до них. На цьому рівні компонентами системної архітектури ОБС є пакети даних, а не комп'ютерні системи. Двома основними елементами цієї схеми є два «сховища» даних – БНД і БД СОБС та БНД і БД СБД. Зупинимось на кожному з них.

На СОБС розташовані так звані «рідні» БНД та БД, тобто ті, які беруть постійну участь лише у процесі надання ОБС, і ними користується лише система ОБС або її комунікаційні ресурси. Оскільки, як вже зазначалося, СОБС та АБІС не мають прямого фізичного поєднання, до «рідних» БНД та БД не можуть належати ті з них, із якими працює АБІС. Крім того, до «рідних» БНД та БД можуть належати лише ті, які не потребують виключного захисту. Призначення цих БНД та БД – забезпечити роботу системи ОБС та/або її комунікаційних ресурсів. Доцільність розташування їх на СОБС пояснюється досить просто. Розміщення на одному фізичному сервері системи ОБС та «рідних» БНД та БД набагато збільшує швидкість роботи системи ОБС та заощаджує сільовий трафік. Якщо б за всіма даними системі ОБС довелося звертатись до СБД, канал зв'язку між ними був би постійно переповненим, а клієнт отримував би потрібні дані з великим запізненням, про онлайн-взаємодію у такому випадку можна було б забути. До того ж, політику безпеки СБД прийшлося б значно пом'якшити, оскільки система ОБС іноді потребує досить великих привілеїв для доступу до деяких даних. Тому розміщення робочих БНД та БД на СОБС – це не лише доцільний, а й необхідний крок.

На СБД розміщуються інші БНД та БД – «електронний фонд бібліотеки». СБД є сховищем документних ресурсів ОБС, тобто тих БНД та БД, які призначені задовольняти документні потреби користувачів. Доцільність збереження цих БНД та БД на окремому фізичному сервері вже обговорювалася, тому тут це питання детально висвітлюватись не буде.

Розглянемо процес адміністрування БНД та БД обох серверів.

СОБС є виділеним сервером, що не поєднаний з АБІС або іншою локальною мережею бібліотеки. Тобто не передбачається можливість адміністрування роботи цього сервера з робочих станцій однієї з локальних мереж бібліотеки. Проте СОБС сам може являти собою серверну локальну мережу, у такому випадку віддалене адміністрування СОБС є можливим. Але тут розглядається простіший варіант – коли СОБС являє собою єдиний

комп'ютер (тобто варіант, коли один логічний сервер є одним фізичним сервером). У цьому випадку адміністрування БНД та БД цього серверу проходить вручну, тобто з терміналу СОБС. Передбачається, що обсяги БНД та БД СОБС є порівняно невеликими, тому їх адміністрування з терміналу (ручне введення даних та/або перенесення їх із фізичних носіїв: дискет, компакт-дисків тощо) не повинно бути складним для бібліотеки.

Адміністрування БНД та БД СБД відбувається по-іншому. Саме на цьому сервері зберігаються всі основні документні ресурси бібліотеки: електронні каталоги та картотеки бібліотеки, повнотекстові БД, Інтернет-архіви, БД мультимедіа інформації, довідкові, фактографічні та інші БНД та БД. Обсяги цих БНД та БД є дуже великими, тому, звісно, ефективно адмініструвати їх вручну майже не можливо. БНД та БД СБД адмініструє АБІС. АБІС – це розгалужена локальна мережа, яка охоплює більшість відділів бібліотеки. Кожен із цих відділів має свої робочі станції АБІС і займається адмініструванням своїх БНД та БД. Відділ каталогізації підтримує електронний каталог, довідково-бібліографічний відділ поповнює довідкові БД, спеціалізовані відділи адмініструють спеціалізовані БНД та БД тощо. АБІС може включати в себе десятки, а то й сотні комп'ютерних систем, копіювального, комутаційного та іншого обладнання. Тобто над наповненням БНД та БД СБД працюють практично всі відділи бібліотеки. Але конкретні механізми роботи АБІС, шляхи її наповнення та адміністрування – дуже широка тема, яка тут детально не розглядається. Для нас важливими є лише два фактори: те, що адміністратором БНД та БД СБД є АБІС, тобто система напряду не зайнята у процесі реалізації ОБС, і те, що ця система є достатньо потужною та розгалуженою для ефективного адміністрування всіх БНД та БД СБД.

Слід також зауважити, що кінцевий користувач інформації отримує доступ до потрібних йому БНД та БД СОБС і СБД по-різному. Як вже зазначалося, від імені користувача у процесі реалізації ОБС виступає КОБС. Доступ КОБС до БНД та БД СОБС на цьому рівні проходить у два етапи. На першому етапі КОБС отримує доступ до системи ОБС, яка тут виконує роль своєрідного «комутатора» доступу. Далі вже система ОБС звертається у якості агента до потрібної БНД або БД СОБС і надає КОБС потрібну інформацію.

Доступ КОБС до БНД та БД СБД проходить у три етапи. КОБС отримує доступ до системи ОБС, яка, у свою чергу, у якості агента має доступ до шлюзу *HTTP-Z 39.50* – клієнта СБД. Така схема доступу не лише забезпечує захист даних на СБД від несанкціонованого доступу користувачів, але й суттєво покращує швидкість роботи СБД, оскільки маршрутизація запитів проходить безпосередньо на СОБС. СБД працює одразу з багатьма БНД та БД, тому функцією шлюзу є уніфікація до-

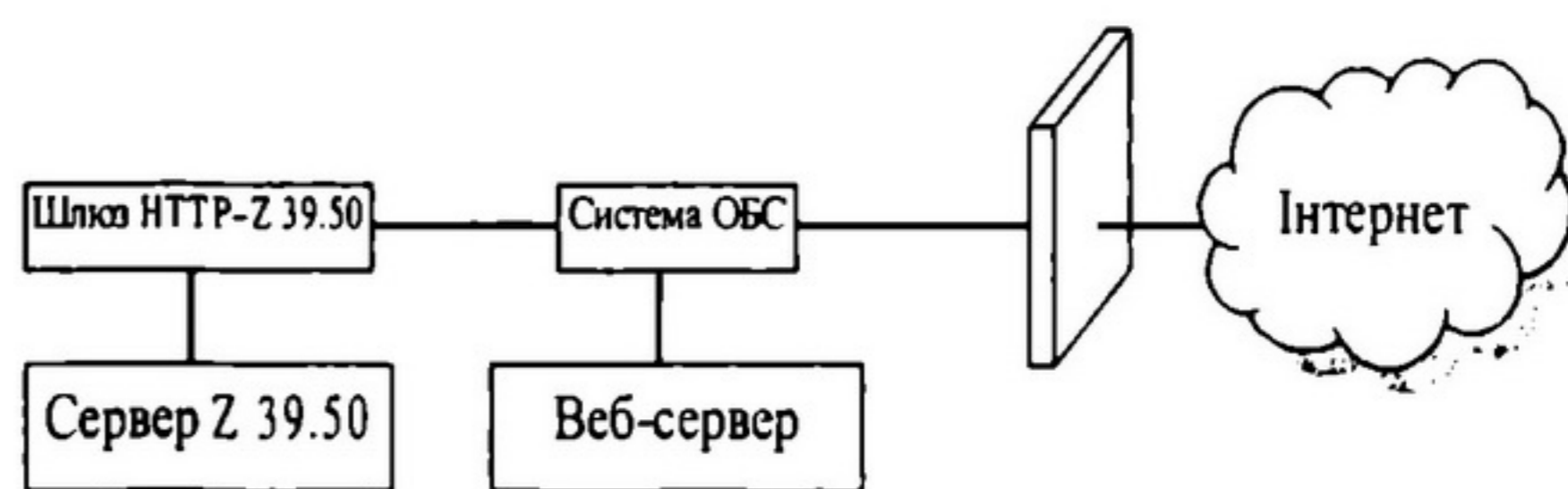


Схема 4. Програмний рівень системної архітектури ОБС

ступу до будь-якої з БД СБД згідно з протоколом *Z 39.50*<sup>6</sup>.

**Програмний рівень** відображає структуру програмної взаємодії у процесі реалізації ОБС (тобто тієї взаємодії, яка відбувається між програмними процесами). Отже, компонентами системної архітектури тут є програми, що виконуються на серверах та комп'ютерах клієнтів. На схемі 4 відображається взаємодія між ними.

Як видно зі схеми 4, на програмному рівні у процесі реалізації ОБС взаємодіють чотири основні програмні компоненти: система ОБС, веб-сервер, програмний шлюз *HTTP-Z 39.50* та сервер *Z 39.50*. П'ятим програмним компонентом є брандмауер (програмний брандмауер або програмне забезпечення апаратного брандмауера).

Тобто, якщо виключити брандмауер, маємо два сервери, які працюють у двох різних протоколах – *HTTP* та *Z 39.50* (питання підтримки веб-сервером інших протоколів: *FTP*, *SMTP*, *IMAP*, *Telnet* тощо – тут ми не розглядаємо).

Система ОБС, веб-сервер та шлюз *HTTP-Z 39.50* розташовані на СОБС. На програмному рівні КОБС напряду звертається до системи ОБС, яка і займається реалізацією ОБС. Залежно від того, який сервіс вимагає КОБС, система ОБС може звернутись до тих програмних компонентів, які містяться на СОБС, або до тих, які містяться на СБД. До тих програмних засобів, які містяться на СОБС, система ОБС може звертатись напряду через веб-сервер (*IIS*, *Apache*, *Xitami* тощо), що проводить її хостинг. На СОБС можуть міститись програмні додатки та програмні бібліотеки системи ОБС. Програмними додатками можуть бути *CGI*-програми, *Java*-класи, *NET*-компоненти тощо.

До програмних компонентів, які містяться на СБД, система ОБС звертається за допомогою шлюзу *HTTP-Z 39.50*. Цей шлюз у якості клієнта *Z 39.50* може звертатись до серверу *Z 39.50*, який розташований на СБД.

Весь процес реалізації ОБС проходить повністю у багаторівневій архітектурі «клієнт/сервер». Тобто кожен із програмних додатків має клієнтську та серверну частини. Спочатку КОБС звертається до серверної частини системи ОБС. Клієнтська частина системи ОБС ре-

<sup>6</sup> Детальніше про це див.: Жижимов О. Л. Введение в *Z 39.50*. – Изд. 2-е, перераб. и доп. – Новосибирск: Изд-во НГОНБ, 2001. – 253 с.

гує зверненням до веб-сервера або серверної частини шлюзу HTTP-Z 39.50. Клієнтська частина шлюзу звертається до серверу Z 39.50, а сервери у якості клієнтів звертаються до програмних компонентів або БД, що запитувались КОБС.

Отже, було розглянуто три рівні реалізації ОБС та проаналізовано, які дії відбуваються на кожному з цих рівнів та структурні компоненти системної архітектури кожного з рівнів. Це вертикальна структура ОБС, горизонтальною структурою ОБС є її поділ на підсистеми, кожна з яких відповідає за реалізацію певного кола послуг, що надаються системою ОБС.

### Характеристика підсистем системної архітектури ОБС

Горизонтальний поділ структурної моделі ОБС – поділ на підсистеми – відповідає переліку тих сервісів, які бібліотека надає в онлайн-режимі. Система ОБС може включати в себе безліч сервісів: пошукові сервери, засоби навігації, електронні енциклопедії, довідники, словники, віртуальні магазини, форуми, чати, телеконференції тощо. За функціонально-цільовою ознакою їх варто об'єднати у певну кількість підсистем, які належать до програмної інфраструктури і оформлені у вигляді програмних модулів.

Програмний модуль – це програма або окрема її функціональна частина, що розглядається як одне ціле у контексті збереження, заміни, трансляції, об'єднання з іншими програмними модулями і її завантаження в оперативну пам'ять ЕОМ<sup>7</sup>. Отже, тут маємо на увазі, що при розробці системи ОБС повинно застосовуватись модульне програмування, яке є ефективнішим при розробці великих та складних програмних систем.

Програмний модуль системи ОБС – це певна сукупність (один або декілька) програмних «пакетів» (assembly), яка відповідає за надання ОБС певного типу та виду. Отже, залежно від профілю бібліотеки та прийняття нею рішення про доцільність надання тих чи інших сервісів, кількість модулів може бути більшою або меншою. Так, якщо бібліотека вважає за необхідне надавати торговельні послуги користувачам, вона створює на сервері торговельний модуль, який може включати в себе інтерактивний магазин та/або аукціон, якщо бібліотека бажає надавати користувачам онлайн-доступ до своїх довідкових сервісів (електронних енциклопедій, інтерактивних словників та ін.), вона створює довідковий модуль тощо. Проте існує чотири обов'язкові модулі, які впливають на роботу всієї системи загалом і тому є базовими.

Обов'язковими модулями є:

1. Модуль доменної маршрутизації (DNS-сервер) – це система, основним призначенням якої є перетворен-

ня доменних імен пристроїв у IP-адреси, або навпаки – IP-адрес у доменні імена. Основою DNS є розподілена ієрархічна база даних<sup>8</sup>.

2. Статистичний модуль – система, призначенням якої є збір, збереження та аналіз кількісних показників роботи серверів, а також користувачів ОБС.

3. Модуль оптимізації – система, призначенням якої є покращання швидкодії серверів, шляхом вибору оптимального (найкращого з можливих) варіанта фізичного розміщення даних.

4. Пошуковий модуль (пошуковий сервер) – комплекс засобів, призначених для знаходження та отримання з документів даних, що мають певні властивості, вказані у запиті.

Крім того, система ОБС може включати в себе й інші модулі<sup>9</sup>:

- ▼ лінгвістичний;
- ▼ довідковий;
- ▼ навчальний;
- ▼ навігаційний;
- ▼ торговельний;
- ▼ комунікаційний.

Усі модулі складаються з програмних «пакетів», кількість яких може бути більшою або меншою залежно від складності модуля.

Програмний «пакет» – це сукупність програмних додатків, бібліотек, гіпертекстових сторінок та інших програмних компонентів, які працюють під єдиним програмним ядром та виконують єдині комплексні завдання.

До кожного «пакета» входить певна кількість агентів, які беруть на себе обов'язки щодо виконання запиту КОБС. Використання технології багатоагентних систем забезпечує швидкісне та якісне виконання запитів модулем системи ОБС. Крім агентів, до їх складу входять скремблери – програми, які збирають та шифрують інформацію для використання іншими програмами модуля. Обов'язковим компонентом кожного «пакета» є журнал, у якому фіксуються дані по роботі з КОБС.

Модулі працюють з БД та/або БнД, які оптимізуються скремблерами або не оптимізуються взагалі. Деякі БнД та БД можуть використовуватись кількома модулями (БД тезауруса може використовуватись пошуковим та лінгвістичним модулем, БД адрес URL – модулем доменної маршрутизації та пошуковим модулем). Усі модулі повинні використовувати глобальні БД, створені для адмі-

<sup>8</sup> DNS-Master. Редактор файлів зон DNS / RU-CENTER [Електронний ресурс]. – Засіб доступу: URL: <http://www.dns-master.ru/help/>. – Загол. з екрана.

<sup>9</sup> Детальніше про це див.: Соловяненко Д. В. Онлайновий бібліотечний сервіс як один з видів бібліотечного сервісу // Наук. праці Національної бібліотеки України імені В. І. Вернадського. Вип. 11 / Відп. ред. О. С. Онищенко. – К., 2003. – С. 201–223.

<sup>7</sup> Воройский Ф. С. Зазначена праця.

ністрування роботи всієї системи ОБС. Прикладом глобальної БД є БД метаінформації про сервер баз даних (СБД): його БнД та БД, типи та схеми даних, пошукові можливості та робочу інформацію. Також усі модулі повинні використовувати єдину БД користувачів СОБС (якщо така БД була створена).

Доступ до СОБС можливий лише під час *сеансу зв'язку*, причому процедури відкриття та закриття сеансу зв'язку повинні суворо контролюватись СОБС. Під час установаження сеансу зв'язку СОБС отримує від клієнта разом із запитом на відкриття сеансу зв'язку певний пакет даних, необхідний для ініціалізації КОБС. Сервером обробляється отриманий пакет даних та створюється *контрольний пакет сеансу зв'язку*. Цей пакет включає:

- ❖ інформацію про користувача ОБС;
- ❖ інформацію про КОБС (його апаратні та програмні можливості, а також його проксі-сервер);
- ❖ журнал сеансу зв'язку (історію запитів, журнал доступу до ресурсів та додаткову інформацію серверів);
- ❖ внутрішні помітки серверу (ідентифікатор пакета, права доступу, тривалість сеансу тощо).

Контрольний пакет поповнюється протягом усього сеансу зв'язку і знищується СОБС лише після закриття сеансу зв'язку. Ця схема забезпечує прозорість доступу до БнД та БД, отже, і легкість контролю за доступом. Якщо відбувається порушення прав доступу, порушник одразу ж ідентифікується сервером за контрольним пакетом. Доступ поза сеансом зв'язку ідентифікується СОБС як абсолютне порушення безпеки серверу, у такому випадку СОБС повинен одразу ж відповісти блокуванням каналу доступу, за яким зафіксоване порушення.

Доцільність використання контрольного пакета сеансу зв'язку важлива ще й з точки зору простоти ведення бібліотечної статистики. Після закриття сеансу контрольний пакет передається на обробку статистичному модулю системи ОБС і лише після цього знищується. Навіть без детальної обробки контрольного пакета модуль може отримати три основні показники бібліотечного сервісу: відвідуваність, кількість читачів та видача. Кількість пакетів, отриманих модулем на обробку – це відвідуваність СОБС. Кількість оригінальних клієнтів – це другий показник (якщо не використовується реєстрація користувачів, то враховується кількість оригі-

нальних хост-систем). Кількість та види наданого сервісу містяться у журналі сеансу зв'язку. Залежно від того, якої складності буде контрольний пакет, статистику можна вести за різною кількістю показників. Такий механізм – простий і одночасно – дуже дієвий. До того ж, інформація з контрольного пакета може бути потрібна також і користувачеві. Тому, за запитом КОБС, перед закриттям сеансу зв'язку, журнал сеансу зв'язку може бути переданий КОБС у вигляді текстового файлу.

Отже підсистеми є відносно незалежними одна від одної. У той же час, якщо не забезпечити координацію між процесами різних підсистем, ОБС втратить цілісність і не зможе розглядатись як комплексний бібліотечний сервіс. ОБС реалізується під час сеансу зв'язку, який має загальносистемний характер. Тобто користувачеві не потрібно відкривати окремі сеанси зв'язку для роботи з різними підсистемами – всі види сервісу можуть бути реалізовані під час єдиного сеансу.

### Висновки

Отже, якщо розглядати ОБС системно, можна побудувати концептуальну модель системної архітектури, що складається з трьох рівнів (апаратного, рівня даних та програмного) та довільної (але не менше чотирьох) кількості підсистем. Кожен із трьох рівнів відображає різний ступінь деталізації моделі. На апаратному рівні взаємодія відбувається між фізичними пристроями, на рівні даних – між пакетами даних, а на програмному – між програмними процесами. Всі підсистеми працюють на всіх рівнях моделі. Як елементи програмної інфраструктури підсистеми оформляються у вигляді програмних модулів. Кожен модуль відповідає за надання певного виду та типу ОБС, він працює відносно незалежно від інших, хоча і може використовувати дані інших модулів. Проте робота деяких модулів має загальносистемний характер. Такі модулі створюються для того, щоб забезпечувати роботу всієї системи (адмініструвати всі її підсистеми), окремі з них повинні бути створені обов'язково. Обов'язковими модулями є модуль доменної маршрутизації, статистичний модуль, модуль оптимізації та пошуковий модуль. Усі інші не є обов'язковими та створюються за бажанням бібліотеки. Отже, системна архітектура ОБС достатньо проста і одночасно достатньо дієва.