

**ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ  
ХАРАКТЕРИСТИК  
СИСТЕМЫ BERGEN INTEGRATION SUITE (BIS)**

## Оглавление

<b>Аннотация.....</b>	<b>3</b>
<b>Термины и сокращения .....</b>	<b>3</b>
<b>Назначение Системы .....</b>	<b>3</b>
<b>Область применения.....</b>	<b>4</b>
<b>Уровни, модули и компоненты Системы .....</b>	<b>6</b>
<b>Основная функциональность Системы .....</b>	<b>7</b>
<b>Принципы построения BIS.....</b>	<b>8</b>
<b>Пользовательский интерфейс BIS Console.....</b>	<b>11</b>

## Аннотация

Настоящий документ содержит описание функциональных характеристик, принципов построения и модульной структуры системы Bergen Integration Suite (далее Система или BIS).

## Термины и сокращения

Сокращение	Полное наименование
BIS	Bergen Integration Suite
ПО	Программа или множество программ, используемых для управления компьютером
ОС	Операционная система
ESB	Enterprise Service Bus, сервисная шина предприятия
K8s	Kubernetes, открытое программное обеспечение для автоматизации развёртывания, масштабирования контейнеризированных приложений и управления ими.

## Назначение Системы

Система представляет собой универсальную платформу создания интеграционных решений на основе технологии транзакционной передачи сообщений.

Система может решать следующие задачи, относящиеся к классу задач систем класса ESB:

1. Гарантированная доставка сообщений потребителю в гетерогенной среде в том числе и по плохим каналам связи.
2. Маршрутизация сообщений.
3. Трансформация сообщений.
4. Буферизация сообщений для согласования производительности разных систем-потребителей.

5. Конфигурирование интеграционных контуров для поддержки сложных ИТ ландшафтов, состоящих из многокомпонентных приложений. Причем все средства конфигурирования находятся «внутри» системы.
6. Наличие средств защиты от ошибок администратора.
7. Развертывание и масштабирование объектов системы на платформе k8s.
8. Централизованное управление территориально распределенной конфигурацией и развертыванием.
9. Мониторинг состояния всех компонентов системы.

Система может являться ядром корпоративной архитектуры, обеспечивая как связь между прикладными системами компании, так и самостоятельное исполнение интеграционных процессов. Например, гарантированную доставку сообщений адресатам по сложным правилам с ведением журнала доставки.

Все объекты системы горизонтально масштабируемы, что дает возможность использовать Систему в высоконагруженных решениях.

### Область применения

Система может применяться в любой области где необходимо:

1. Надежно связать приложения между собой путем обмена сообщениями, в том числе и по плохим каналам.
2. Выстроить процессы гарантированной доставки, маршрутизации и трансформации сообщений.
3. Отслеживать этапы исполнения процессов в том числе для контроля бизнес-метрик и постоянной оптимизации процессов.
4. Выстроить процесс централизованного учета всех трансформаций и маршрутизаций и поставить под контроль процесс их модификации.

5. Уменьшить число трансформаций, чтобы сделать ИТ-ландшафт организации более простым и управляемым
6. Централизовать вопросы ограничения доступа, авторизации, аутентификации
7. Централизовать аудит взаимодействий.
8. Обеспечить платформу (хостинг) для исполнения прикладных приложений, вызываемых посредством отправки/получения сообщений и масштабируемых посредством увеличения числа экземпляров приложения, при сохранении гарантированной транзакционной доставки сообщений вне зависимости от территориального расположения экземпляров.

Примеры возможного применения:

1. Доставка данных о работе (теле)коммуникационного оборудования.
2. Сбор и агрегация данных системы оповещения об инцидентах по типу ЭРА\ГЛОНАСС.
3. Сбор и агрегация данных от датчиков IoT, например, о состоянии посевов зерновых или бытовых датчиков расхода воды.
4. Сбор и агрегация данных от датчиков GPS/ГЛОНАСС от многочисленных объектов, например, вагонов, грузовиков или курьеров.
5. Обеспечение сложных правил доставки прикладных сообщений (корреспонденции) до потребителей в сильно территориально распределенной компании.
6. Хостинг сложных микросервисных приложений с гарантированным вызовом приложения со стороны потребителя.
7. Интеграционная Шина масштаба Предприятия (ESB), обеспечивающая связь всех приложений территориально

распределенной организации, при централизации средств контроля, безопасности и аудита.

## Уровни, модули и компоненты Системы

Логически Система разделена на два основных уровня (слоя):

- уровень транспорта (Messaging)
- прикладной уровень (ESB).

Каждый уровень имеет свои специфические объекты, все параметры которых настраиваются исключительно средствами системы.

Связи между уровнями обеспечиваются специальными объектами – «точки подключения» (endpoints).

В системе BIS можно выделить следующие модули:

1. Домен (Domain) – единое логическое объединение объектов Системы и связей между ними.
2. Централизованная консоль управления, включает:
  - a. Интерфейс пользователя
  - b. Базу данных Доменов и справочников
  - c. Сервер приложений, связывающий пользовательский интерфейс и БД.
3. Среда выполнения, которая представляет собой k8s платформу.

Все модули системы связаны между собой. Например, консоль управления обеспечивает развертывание (deploying) на платформе k8s экземпляров (instances) каждого объекта BIS, получает и отображает состояние каждого из экземпляров каждого объекта.

**Домен** логически связывает все объекты системы, например, брокер, его очереди и его акцепторы.

## Основная функциональность Системы

Система позволяет конфигурировать:

1. Транспортный слой (Messaging), который в свою очередь состоит из трех основных конструкций:
  - a. Брокер сообщений
  - b. LB-пара – (live-backup pair) связка ведущего (live) брокера и его резерва (backup). Данная конструкция используется для реализации наиболее высоконадежных (HA) контуров доставки, когда даже кратковременная задержка с доставкой сообщений недопустима, так же как и потеря и дублирование сообщений.
  - c. Кластер брокеров. Позволяет наращивать производительность транспортной системы, однако за счет возможных задержек доставки части сообщений.
2. Слой прикладных сервисов-обработчиков (ESB), в качестве которых могут быть использованы сервисы, написанные на любом языке программирования и реализующие любую функциональность. Совместимость с требованиями системы обеспечивается согласованностью интерфейсов взаимодействия с транспортным уровнем. Объектом прикладного уровня является объект типа «Сервис».
3. Связь между «Сервисом» и уровнем транспортных объектов осуществляется через специальные промежуточные объекты - «точки подключения» (endpoints), которые позволяют изолировать прикладные объекты от подробностей соединения с транспортными объектами.
4. Система позволяет управлять слоем исполнения приложений на платформе k8s, обеспечивая масштабирование любого объекта как транспортного, так и прикладного уровня. При этом, платформа исполнения не может оказать влияния на внутреннюю конфигурацию системы. Платформа выполнения может быть изменена, для этого необходимо лишь изменить шаблоны скриптов развертывания.

5. Система позволяет производить мониторинг в режиме реального времени состояния всех своих объектов.

### Принципы построения BIS.

BIS построена по следующим принципам:

1. Транспортный уровень (Messaging):
  - a. Обмен информацией между любыми узлами системы происходит путем обмена сообщениями, осуществляемого в рамках транспортной транзакции.
  - b. Транспортная транзакция (message transaction) – это единый акт пересылки сообщения между двумя узлами. Пересылка либо завершается целиком до последнего байта, либо целиком не совершается. Для обеспечения транспортной транзакции используется механизм обмена специальными служебными подтверждениями ( acknowledgment).
  - c. Физически транспортная транзакция реализуется минимальным набором базовых библиотек.
  - d. Каждый узел системы, знает весь набор транспортных альтернатив (т.е. узлов контрагентов), ему доступных.
  - e. Если при выполнении транспортной транзакции, произошел сбой, то базовые библиотеки на каждом узле произведут откат транзакции и попытку ее повтора, возможно, с другим транспортным контрагентом. Именно поэтому каждый узел должен знать свой набор возможных транспортных контрагентов.
  - f. Транспортные библиотеки изолируют от прикладного уровня все сложности, связанные с выполнением транспортной транзакции, путем предоставления универсальных методов «Отправить» / «Получить».



- g. Любой процесс в BIS представляет собой последовательность транспортных транзакций.
- h. Поскольку каждая транспортная транзакция обеспечивает гарантированную доставку, их последовательность также обеспечивает гарантированную доставку.
- i. При этом в системе отсутствуют выделенные уникальные компоненты, например такие как балансировщики нагрузки, способные оказаться критичной точкой отказа.
- j. К транспортному уровню относятся следующие объекты BIS:
- i. Брокер (standalong) – отдельный (без средств отказоустойчивости) транспортный узел, позволяющий осуществлять транспортные транзакции со следующими объектами:
    - Очередь – запись, хранение и раздача сообщений по принципу – одно и только одно сообщение одному присоединенному (connected) к очереди клиенту.
    - Топик - запись, хранение и раздача сообщений по принципу – одно и только одно сообщение каждому «присоединенном» (connected) к топикю клиенту.
    - Бридж – псевдо очередь на брокере, указывающая на очередь на другом брокере – мост между брокерами. В остальном поведение бриджа аналогично поведению обычной очереди.
  - ii. Lb-пара (live-backup pair) – Брокер и его резерв (backup) – позволяющие создать отказоустойчивый транспортный узел. В случае падения ведущего (live) брокера, его место в системе занимает резервный (backup) брокер. Причем это происходит «незаметно» для прикладного уровня. Транспортные библиотеки продолжают с резервным

брокером транспортные транзакции, начатые с основным брокером. Для обеспечения этой функциональности основной и резервный брокеры имеют общее хранилище данных, содержащее состояние всех очередей, топиков и бриджей Lb-пары. Live и backup брокеры размещаются на разных нодах k8s.

- iii. Кластер брокеров. Представляет собой объединение нескольких брокеров, что позволяет практически линейно наращивать пропускную способность кластера по сравнению с отдельным брокером. Однако, при падении одного из брокеров кластера, находящиеся на этом брокере сообщения будут недоступны до момента поднятия этого брокера. Платформа эксплуатации BIS – k8s автоматически поднимет упавший брокер, что делает время его недоступности минимальным. Однако, поскольку время недоступности «упавшего» брокера все же не равно нулю, не рекомендуется использовать кластер для высоконагруженных и при этом высокодоступных контуров. В таких случаях вместо кластера следует использовать lb-пару или несколько lb-пар.

## 2. Прикладной уровень (ESB).

- a. Задача этого уровня – выполнение прикладных задач с опорой на средство доставки сообщений. К таким задачам может относиться:
  - i. Маршрутизация. Принятие решения об адресе доставки на основе содержимого сообщения.
  - ii. Трансформация сообщения, например, xslt.
  - iii. Опрос внешней системы для обогащения данных в сообщении.
  - iv. Создание специфических адаптеров к внешним системам.
  - v. Прочие классические задачи ESB.

- b. Важно, что Система позволяет объединять сервисы в последовательность шагов, создавая процессы обработки. Состояние каждого процесса – это состояние сообщения, обрабатываемого в этом процессе. Система не использует каких-либо централизованных процессных обработчиков.
- c. К прикладному уровню относится объект типа «Сервис».

## Пользовательский интерфейс BIS Console

Пользовательский интерфейс позволяет осуществлять конфигурирование Системы и контролировать её текущее состояние и производить мониторинг компонент.

Пользовательский интерфейс представляет собой web-приложение, состоящее из нескольких экранов (закладок). Основные экраны:

1. Логин и проверка прав.
2. Администрирование – основные справочники доменно-зависимые и доменно-независимые.
3. Конфигурирование:
  - a. Уровень транспорта - Messaging
    - i. Брокеры
    - ii. Lb-пары
    - iii. Кластеры
  - b. Прикладной уровень - ESB
    - i. Сервисы
  - c. Связи между обоими основными слоями
    - i. Точки подключения – эти объекты указывают на физический адрес транспортного объекта, например: брокер, акцептор брокера и очередь.

4. Среда выполнения, обеспечивающая развертывание (deploying) объектов приложения в виде набора контейнеров в среде k8s. А также мониторинг состояния каждого объекта.



### Составили

<b>Версия</b>	<b>Дата</b>	<b>Фамилия, имя, отчество</b>	<b>Должность исполнителя</b>	<b>Подпись</b>
0.1	24.02.21	Харитонов А.Н.	Аналитик	
0.2	24.03.21	Харитонов А.Н.	Аналитик	
1.0	31.03.21	Харитонов А.Н.	Аналитик	
1.3	31.03.21	Харитонов А.Н.	Аналитик	