

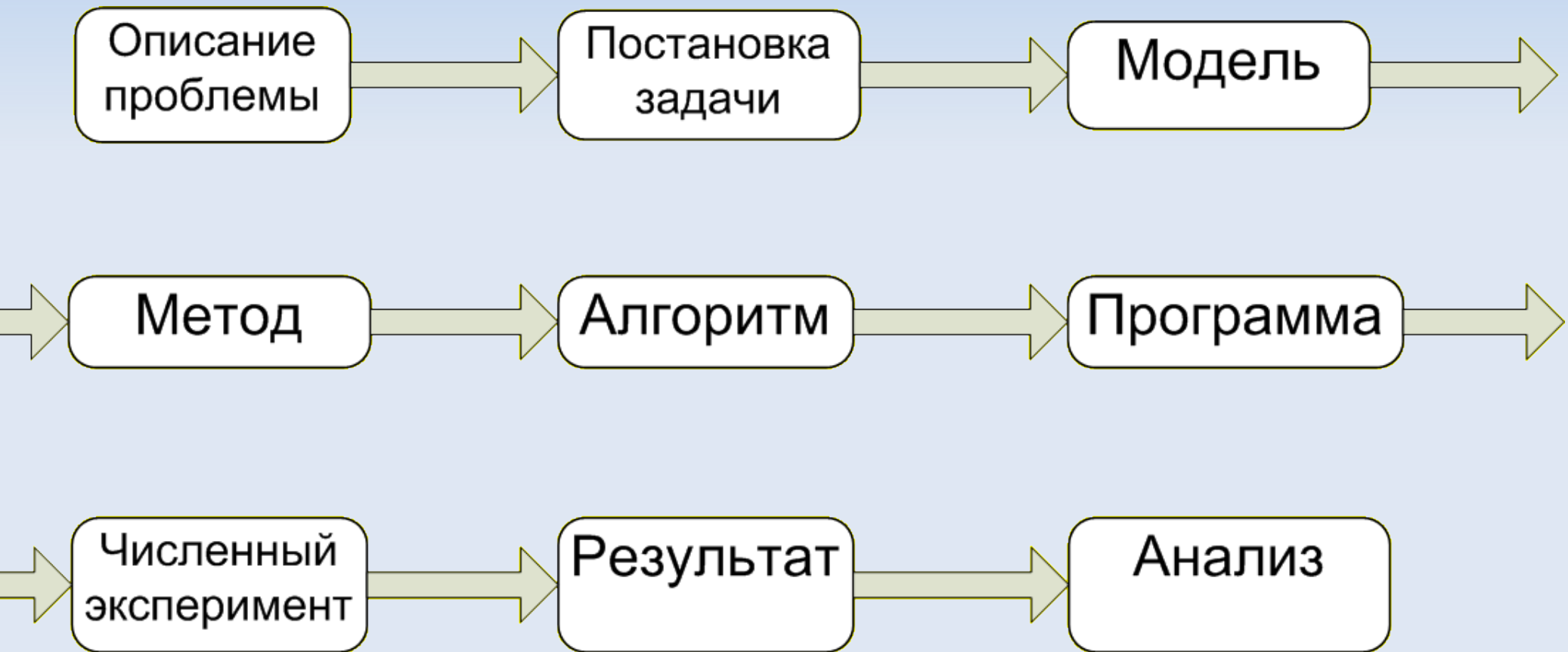
# **Когнитивный процессинг социальных сетей**

Сергей Парамонов, 673ей группы МФТИ.

Научный руководитель: Рыков В.В. к. фил. н.

# Принципиальная схема

В основу работы легла следующая схема:



# Открытая проблема

- Консалтинговое и экспертное сообщество Ovum объявляет конкурс среди студентов по социальным сетям.
- Условия конкурса\*:  
Распространение социальных сетей в последние 3-5 лет выявило как ряд дополнительных преимуществ для компаний в организации бизнеса, так и ряд проблемных моментов и рисков. К последним можно отнести:
  - утрату контроля за прохождением информации вне установленных каналов коммуникации.
  - возможность для конкурирующих компаний сбора информации через социальные сети и позиционирования своей стратегии.
  - ущерб для репутации компании, когда сотрудники обсуждают внутренние проблемы на общественных сайтах или, хуже того, конфиденциальную информацию, связанную с клиентами...

# Постановка задачи

Нахождение компромисса между рисками и преимуществами по контролю и возможностям соц. сетей.

- Сохранить креативные и инновационные возможности использования сетей.
- Формальные критерии:
  - - Каким образом предложения могут устроить консервативно настроенный бизнес?
  - Насколько инновационно предложение в организационном и технологическом плане?
  - Насколько практически осуществимы предлагаемые идеи?
  - Насколько хорошо будут работать предложения в различных отраслях промышленности и разных национальных бизнес - культурах?
  - Насколько сбалансированы предложения между усилением контроля и извлечением пользы от вхождения в социальные сети?

# Модель

Сущности модели:

- Сети\* с заданной на них структурой:
- Операции над сетями
- Метаструктуры сетей

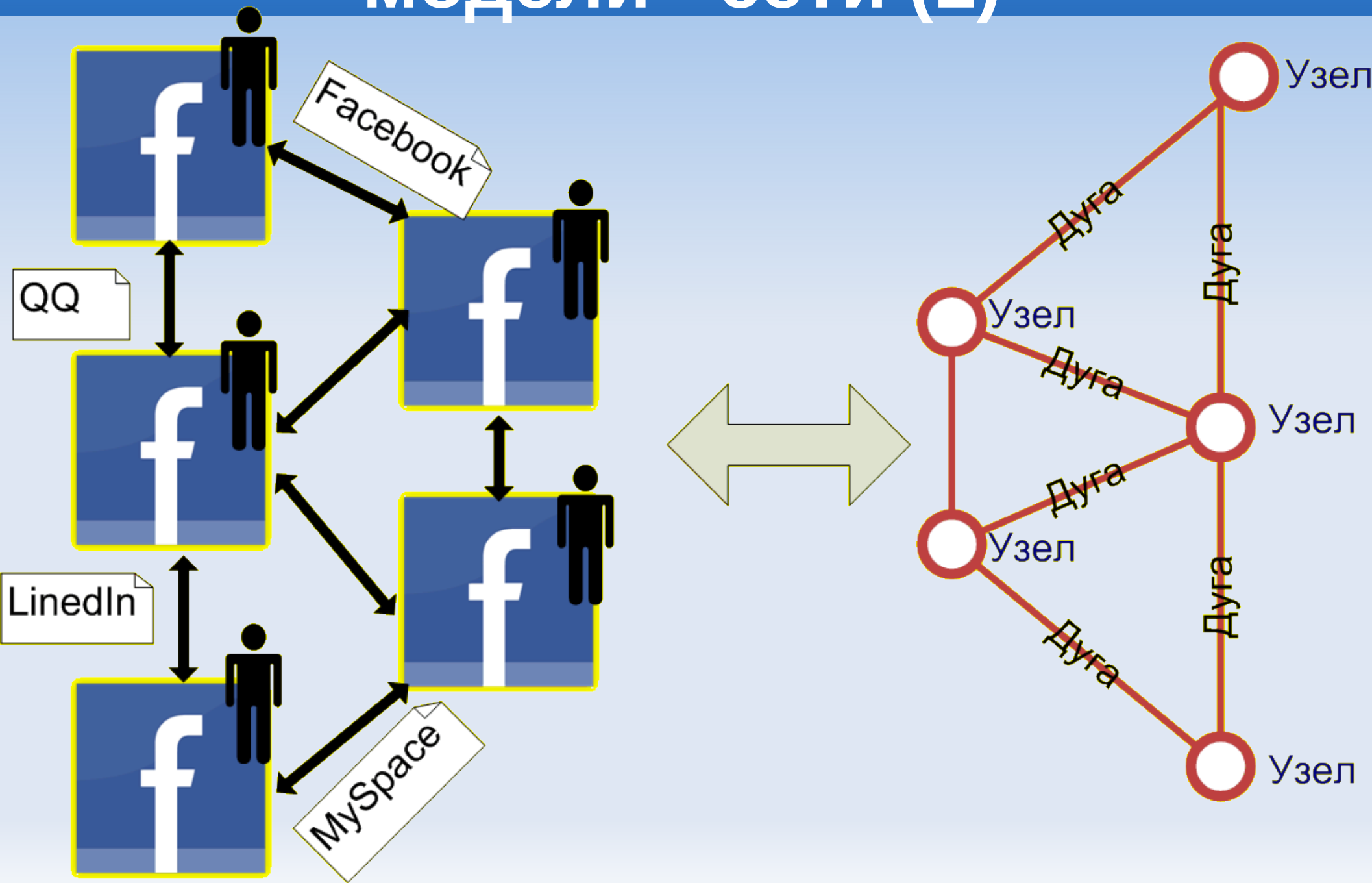
\*Инtranет системы документо-оборота:

*IBM Lotus Notes, Office SharePoint, etc*

Социальные сети в Web: *facebook, mySpace, etc*

Различные вариации CMS: *wiki, google docs*

# Подробнее об элементах модели - сети (1)



# Подробнее об элементах модели - сети (2)

- То есть, можно сказать, что под сетью мы подразумеваем ориентированный граф
- **Важно(!)**, что в нашей модели интранет тоже представим в таком виде
- Значит, что наши элементы обладают свойством замыкания (closure)

# Подробнее об элементах модели – операции(1)

- Так как мы имеем дело с ориентированными графами, то здесь применимы большинство классическим графовых алгоритмов
- DFS – поиск в глубину
- BFS – поиск в ширину
- SPF – алгоритм Дейкстры

Продемонстрируем один из них(BFS).



# Подробнее об элементах модели – операции(2)



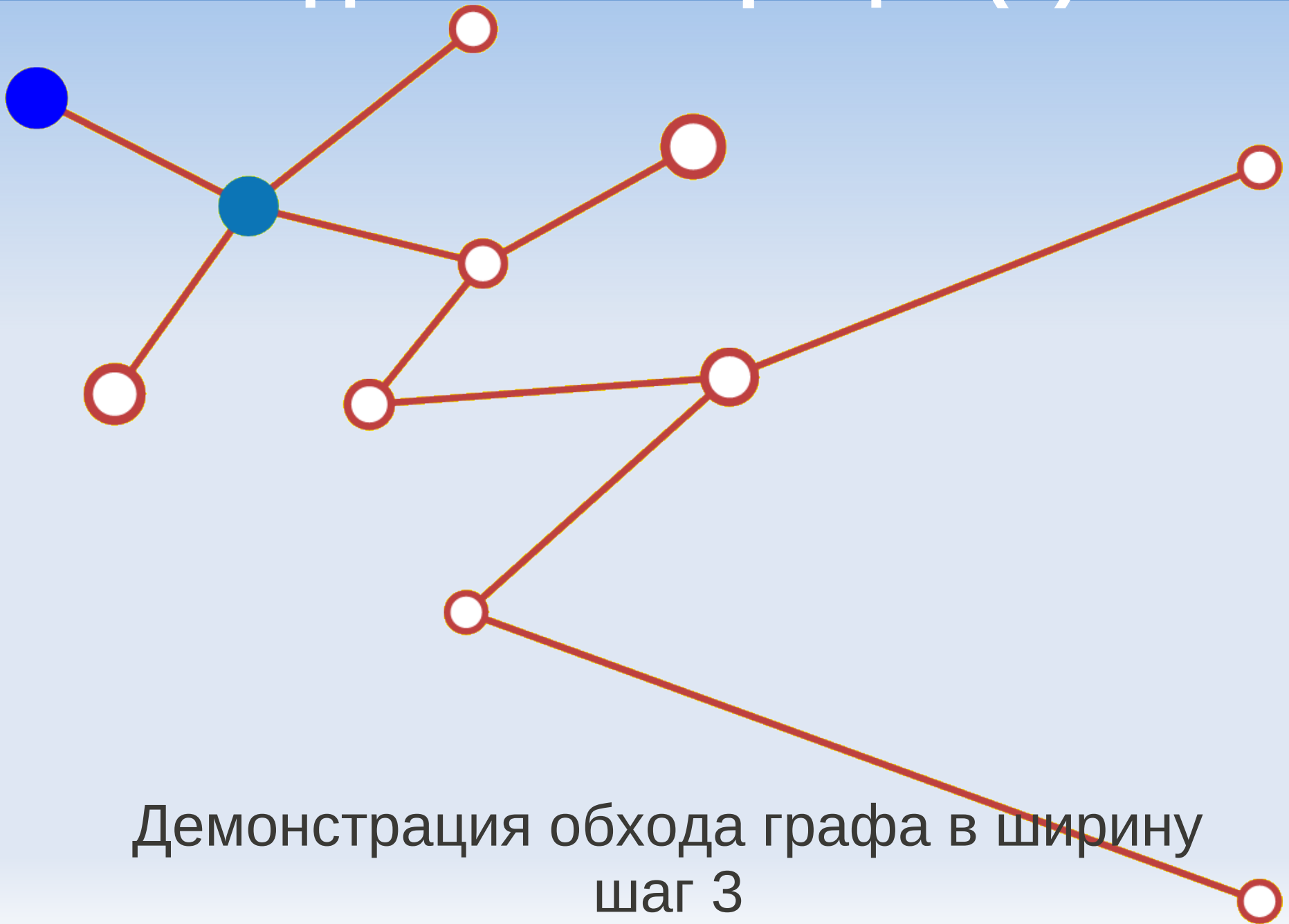
# Демонстрация обхода графа в ширину

# Подробнее об элементах модели – операции(2)

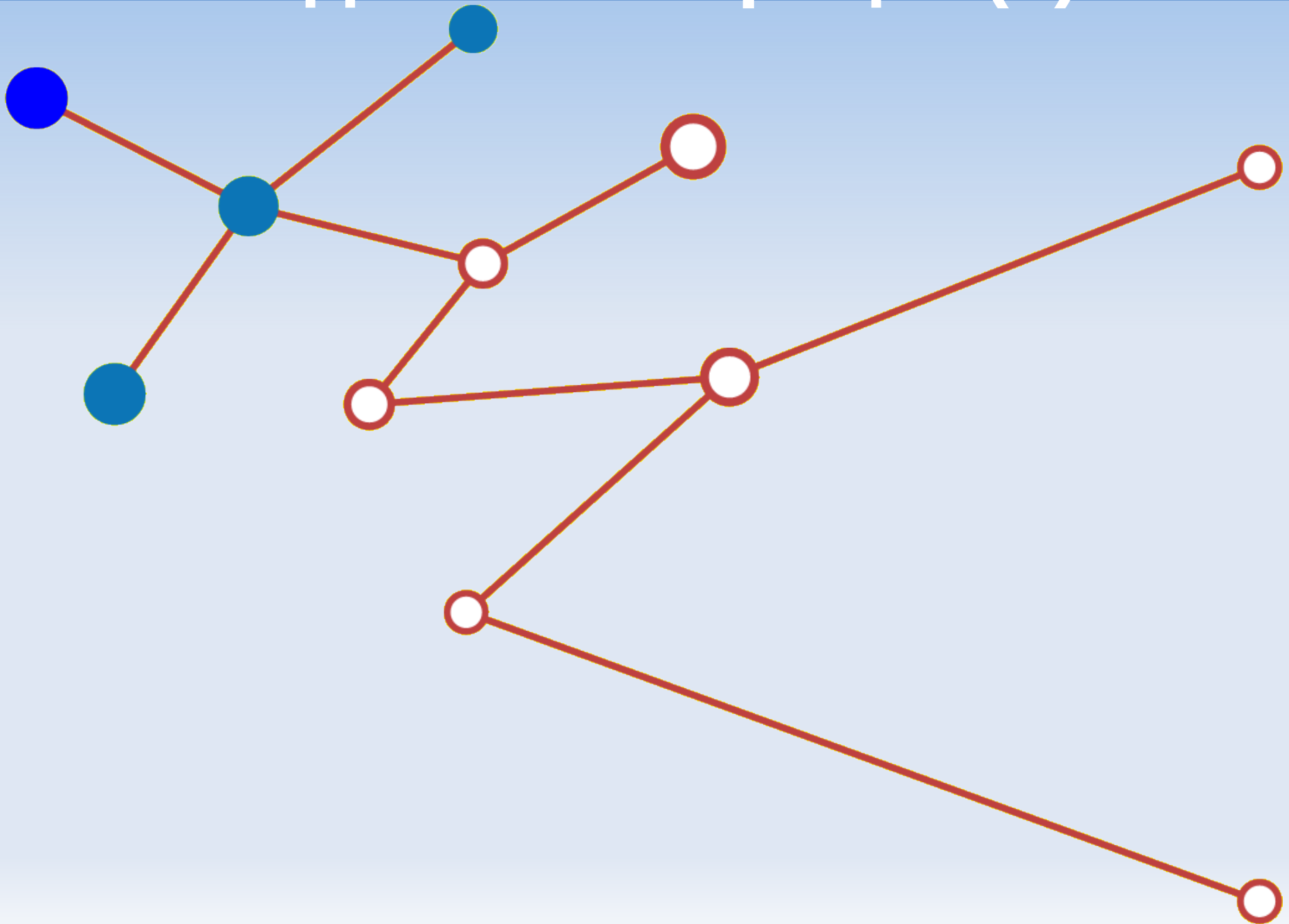


# Демонстрация обхода графа в ширину

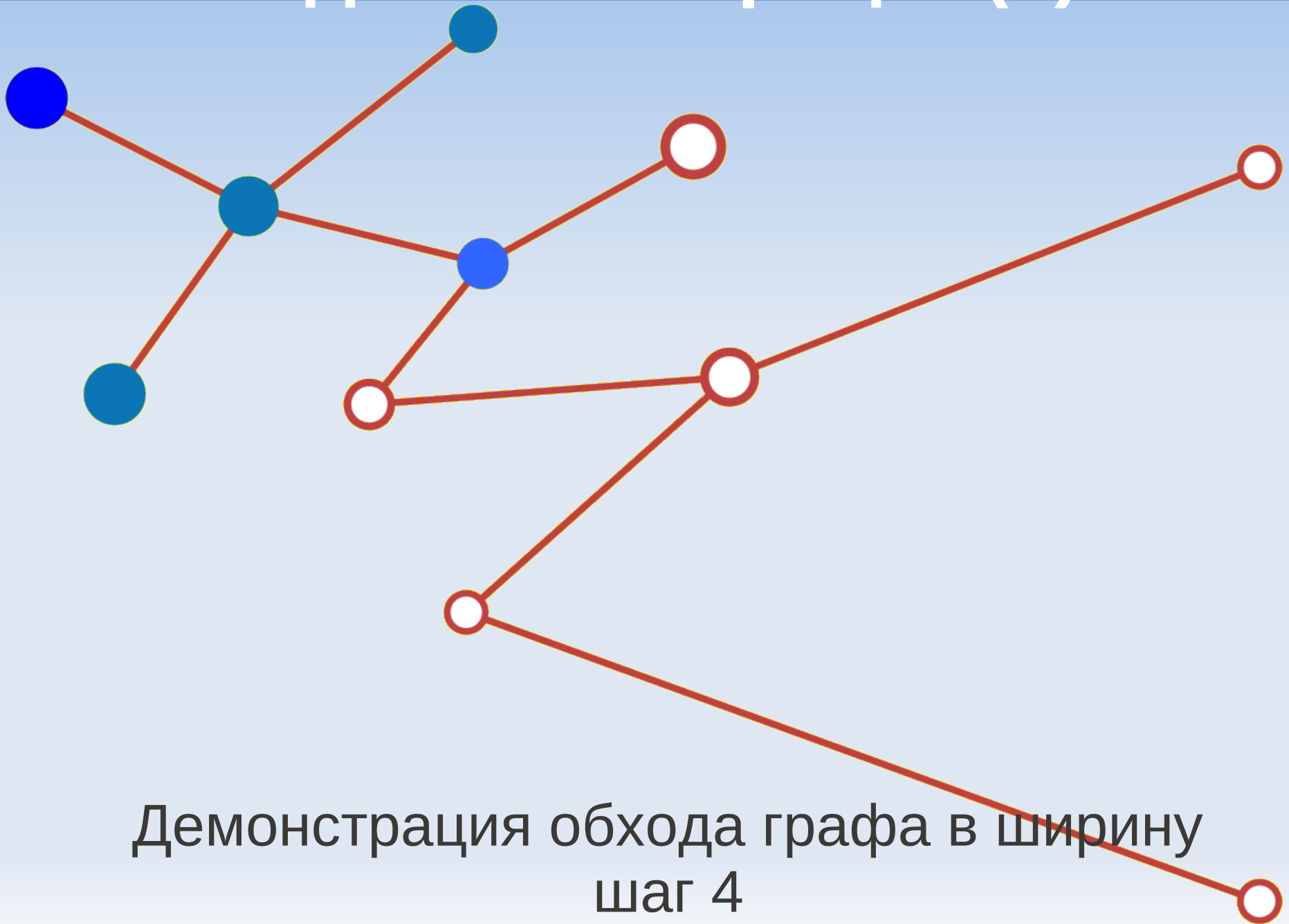
# Подробнее об элементах модели – операции(2)



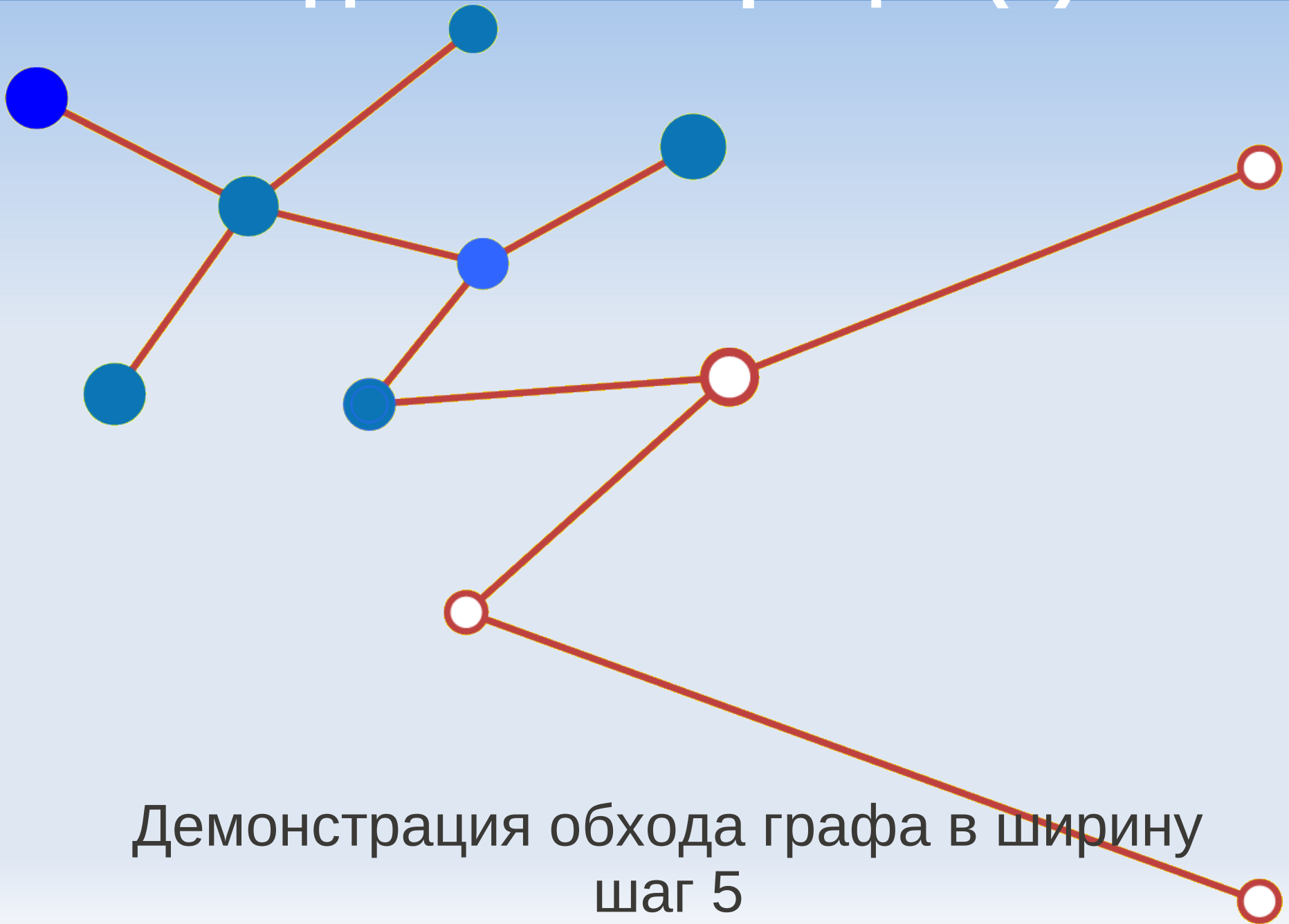
# Подробнее об элементах модели – операции(2)



# Подробнее об элементах модели – операции(2)

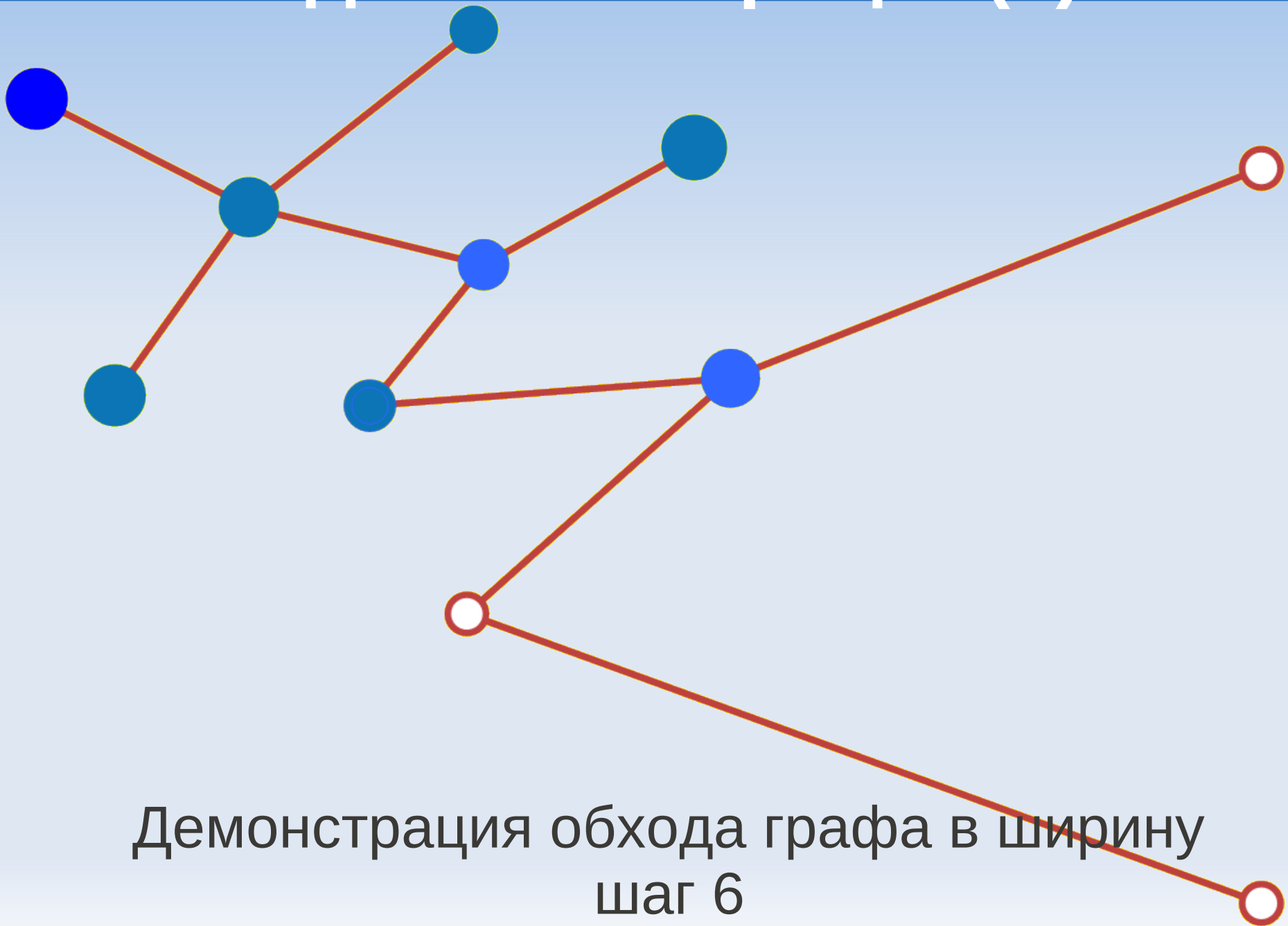


# Подробнее об элементах модели – операции(2)

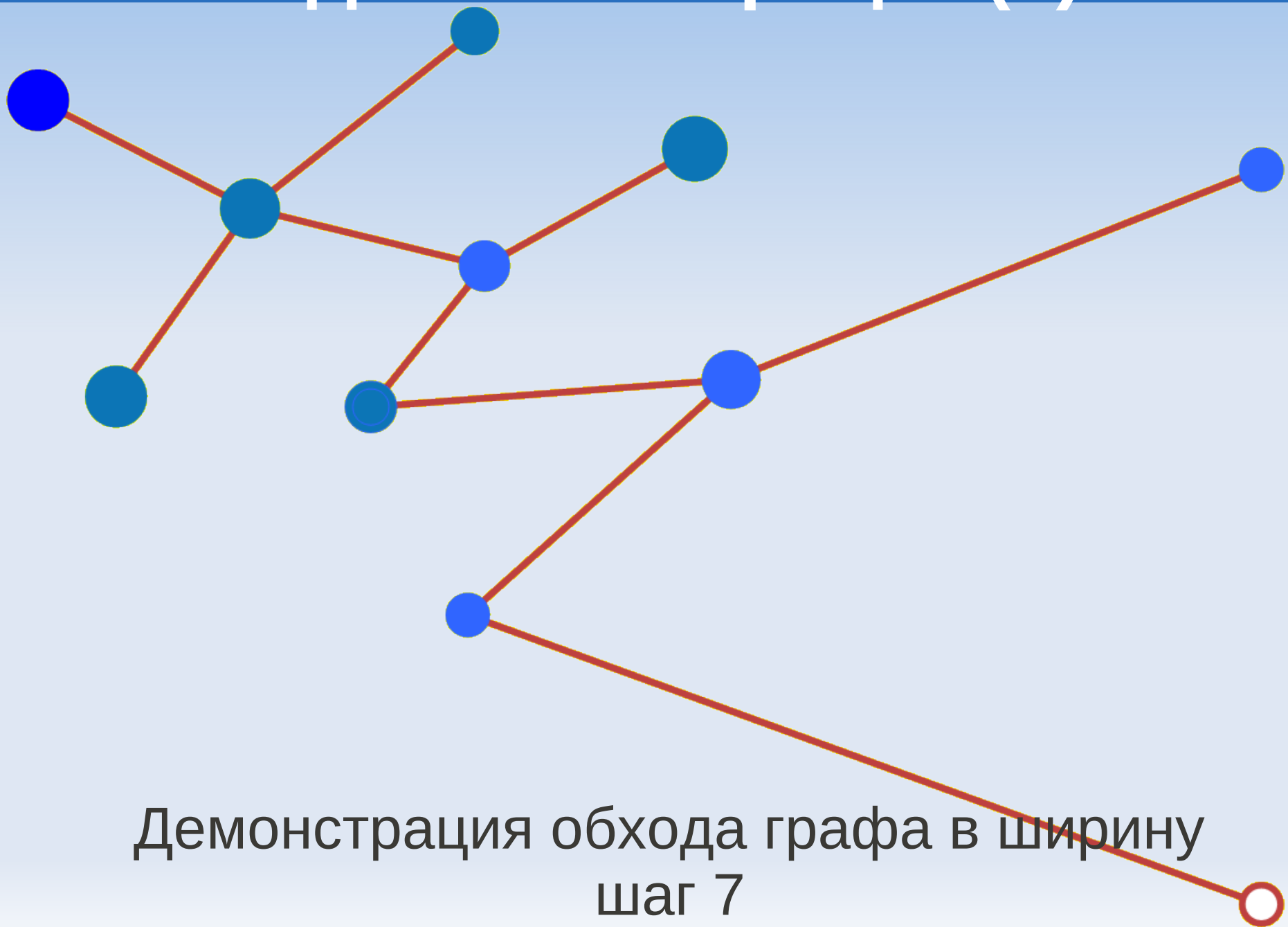


Демонстрация обхода графа в ширину  
шаг 5

# Подробнее об элементах модели – операции(2)



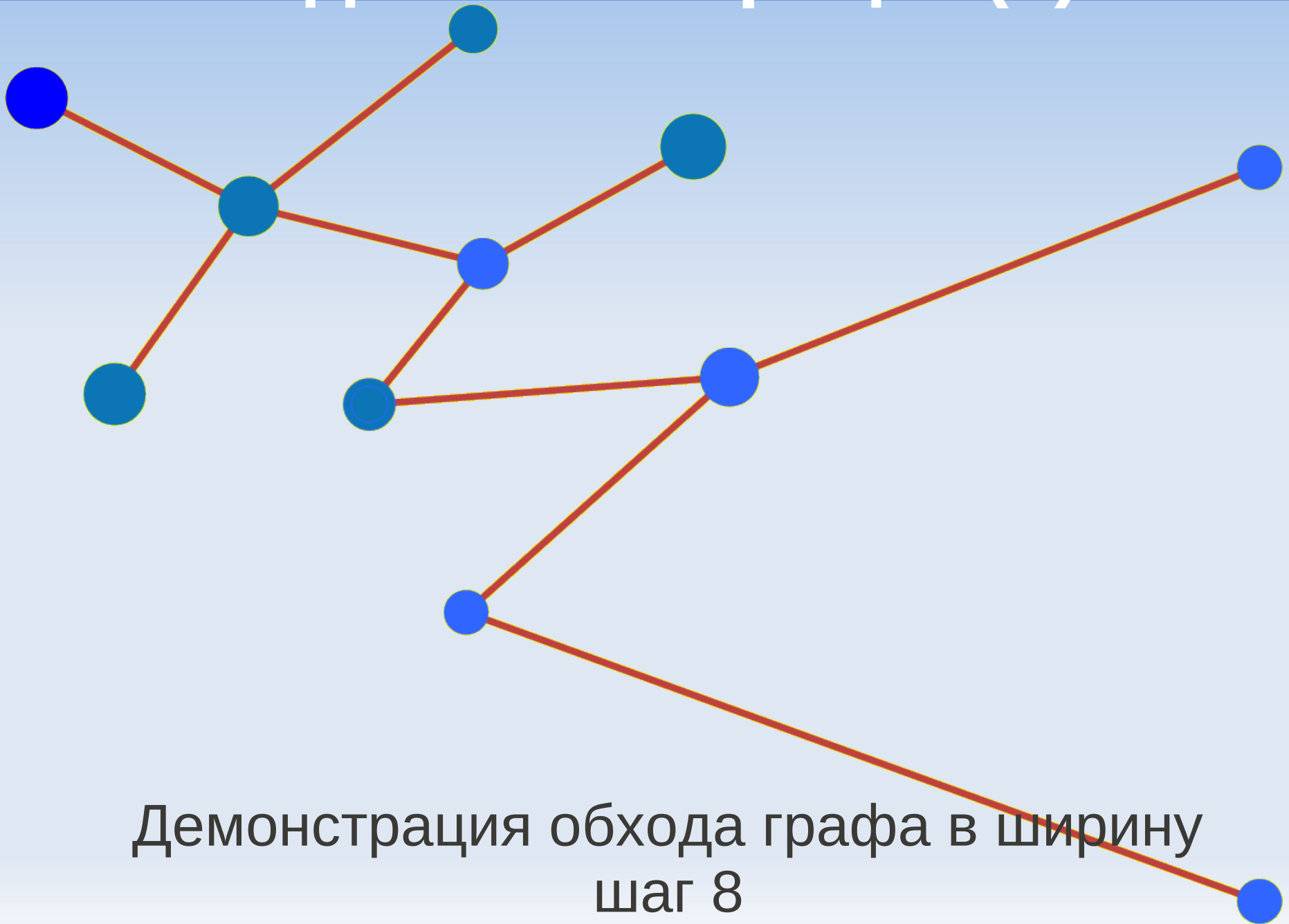
# Подробнее об элементах модели – операции(2)



# Демонстрация обхода графа в ширину шаг 7



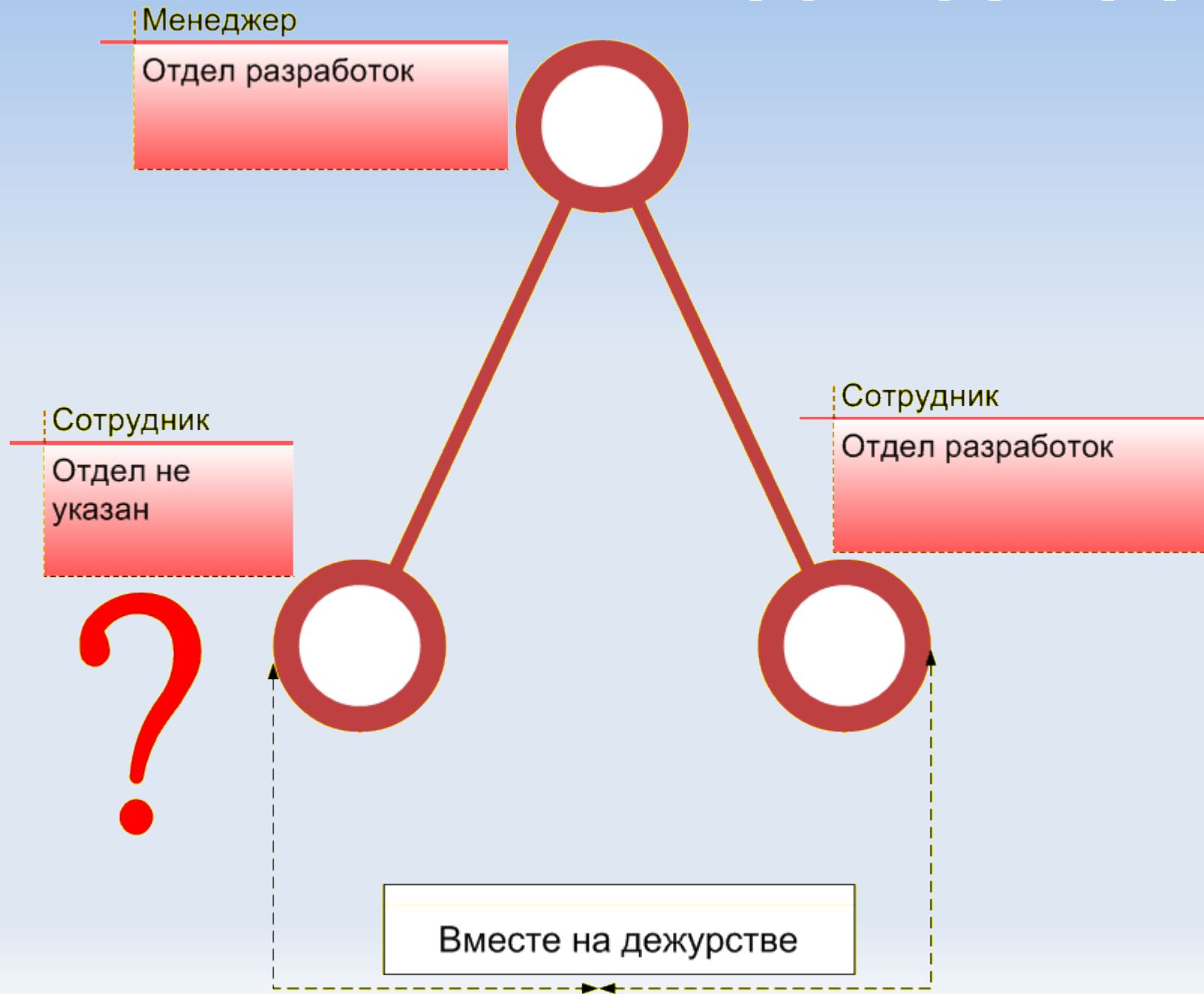
# Подробнее об элементах модели – операции(2)



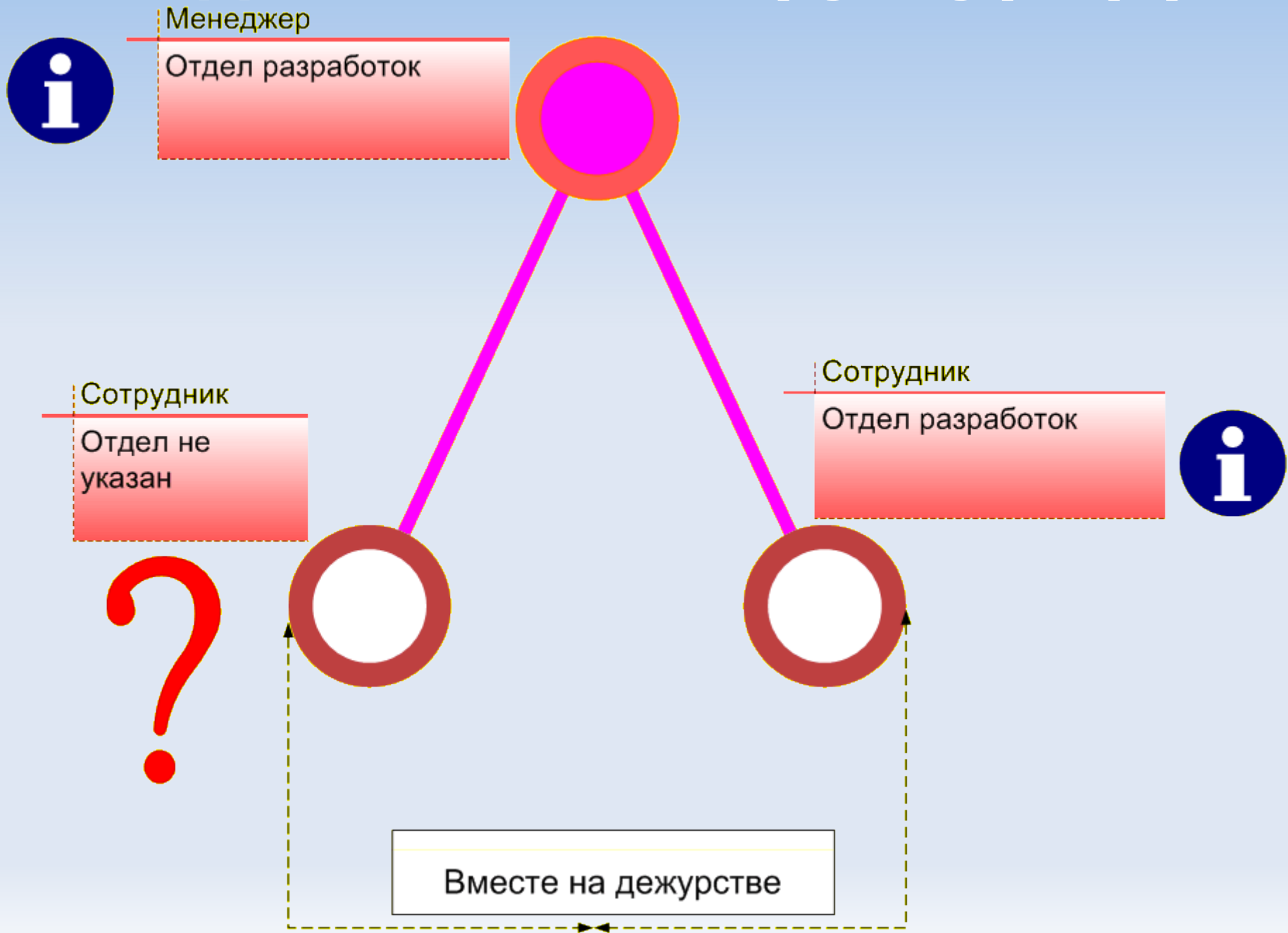
# Подробнее об элементах модели – метаструктура(1)

- Под метаструктурой будем понимать наличие структуры, которая содержала бы в себе описание структур из нашей модели
- Продемонстрируем это простым примером:  
Нам нужно проверить к какому департаменту принадлежит один из работников, учитывая, что явно в его профайле это не указано.

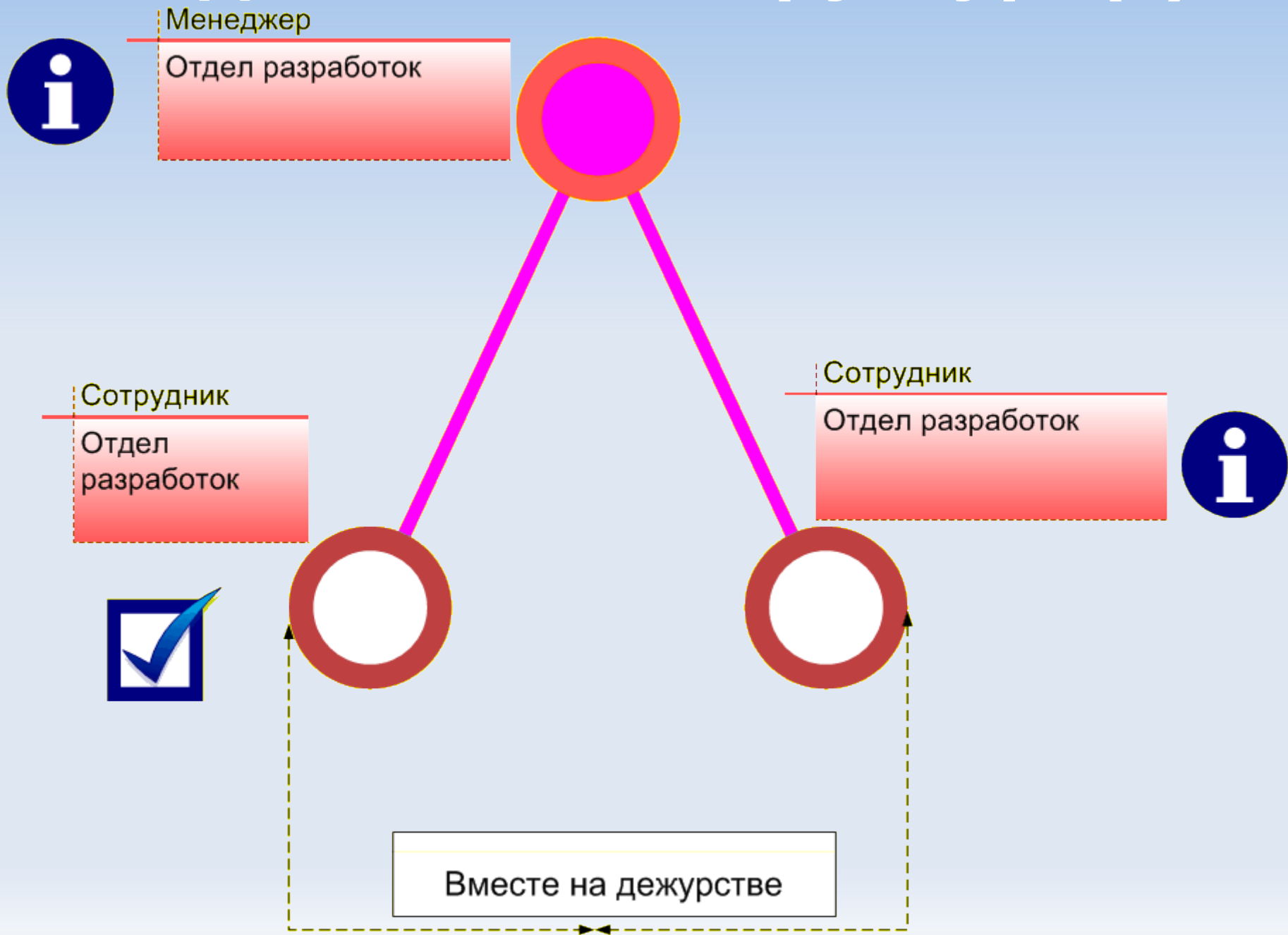
# Подробнее об элементах модели – метаструктура(2)



# Подробнее об элементах модели – метаструктура(3)

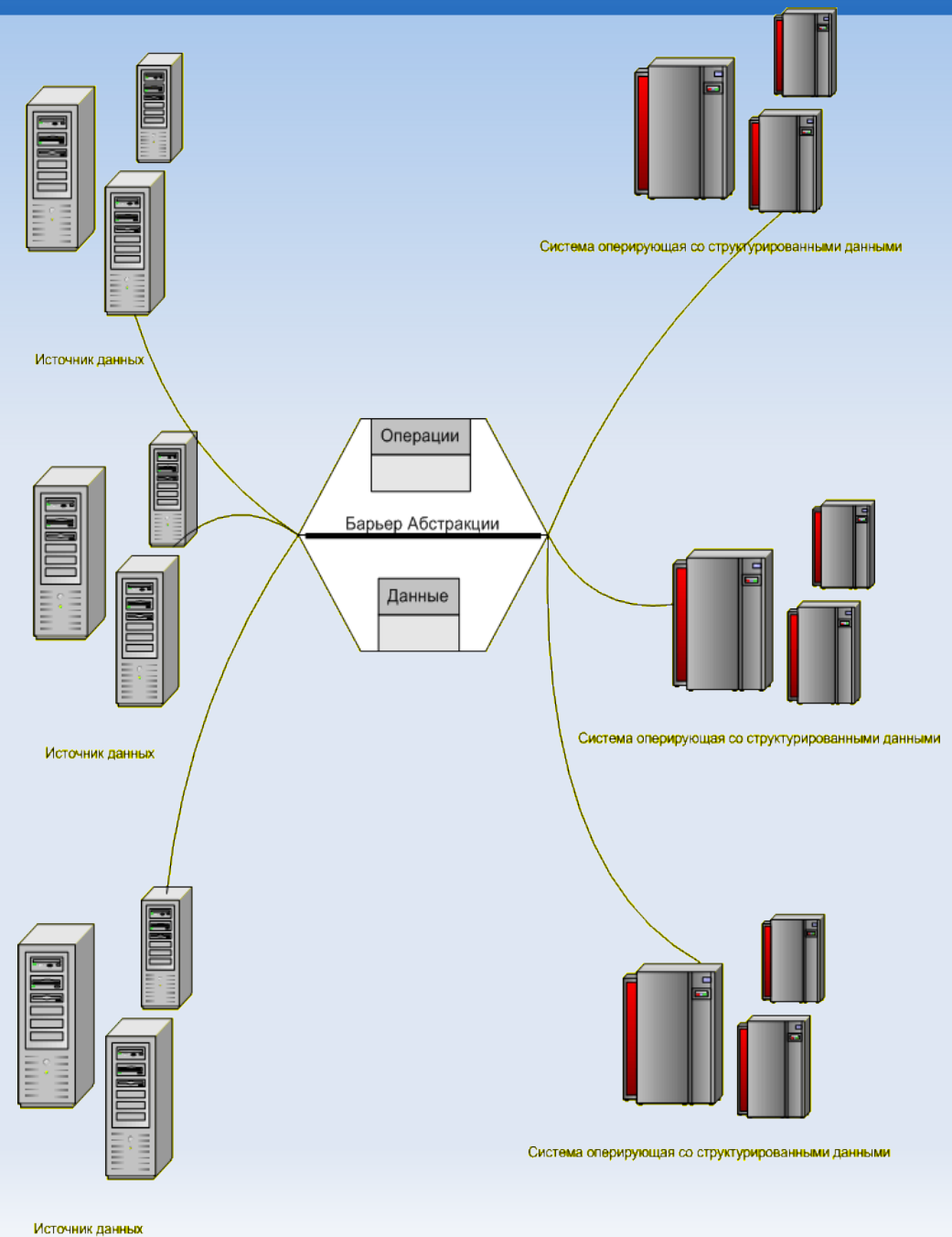


# Подробнее об элементах модели – метаструктура(4)

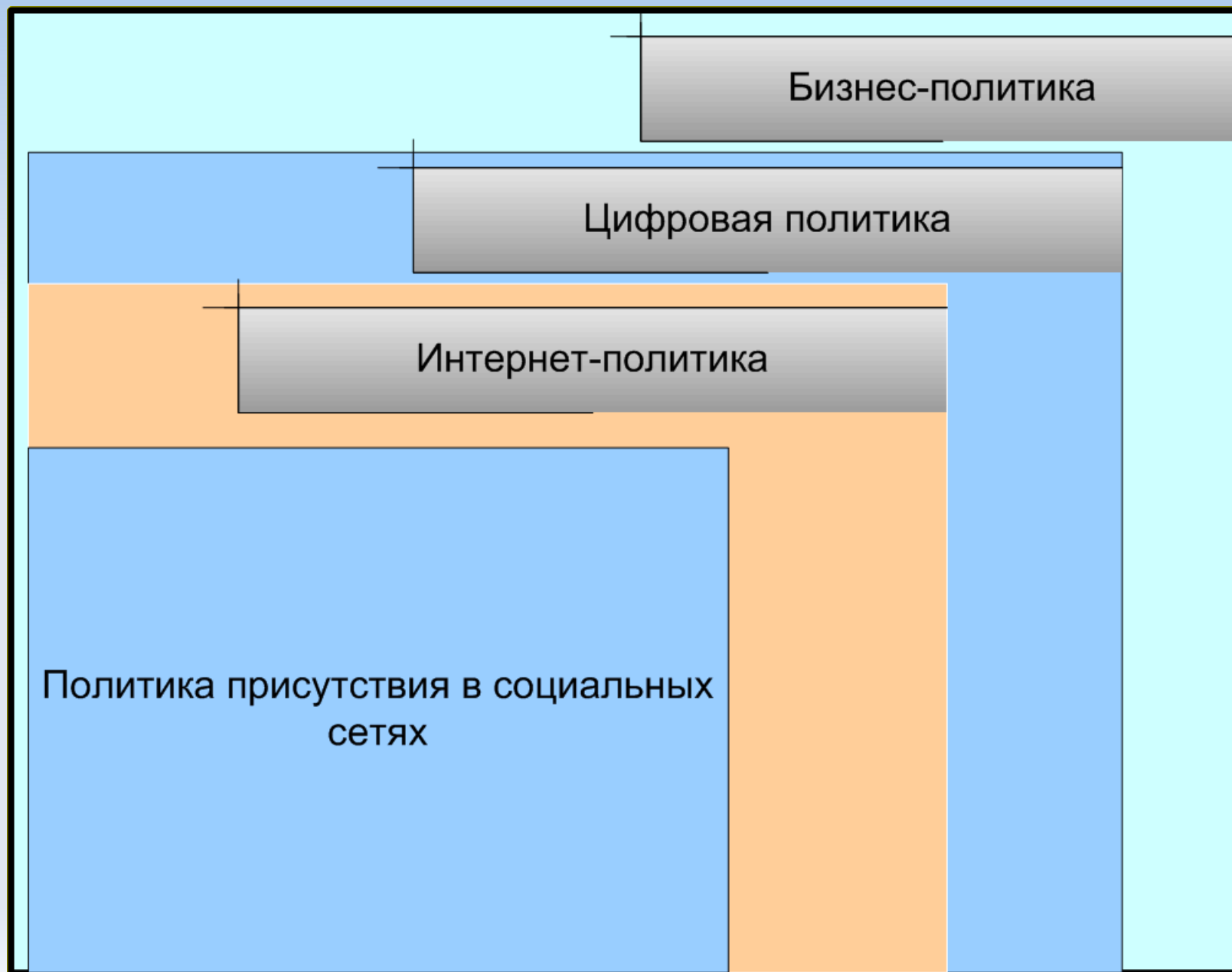


# Алгоритм

- Изменение политики присутствия
- Создание внутренней метаструктуры
- Открытая интеграция



# Алгоритм. Политика доступа.

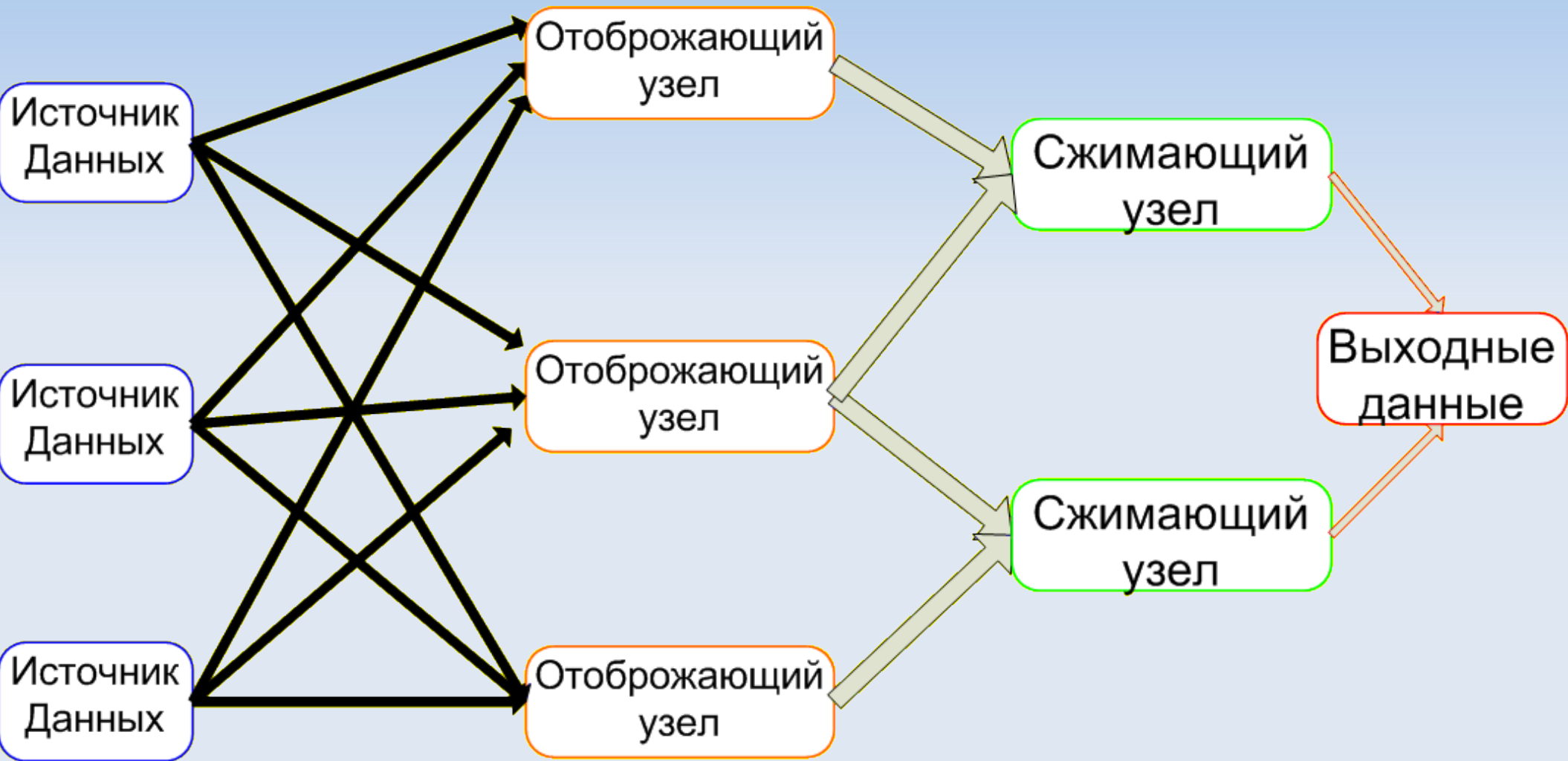


# Алгоритм. Политика доступа(2)

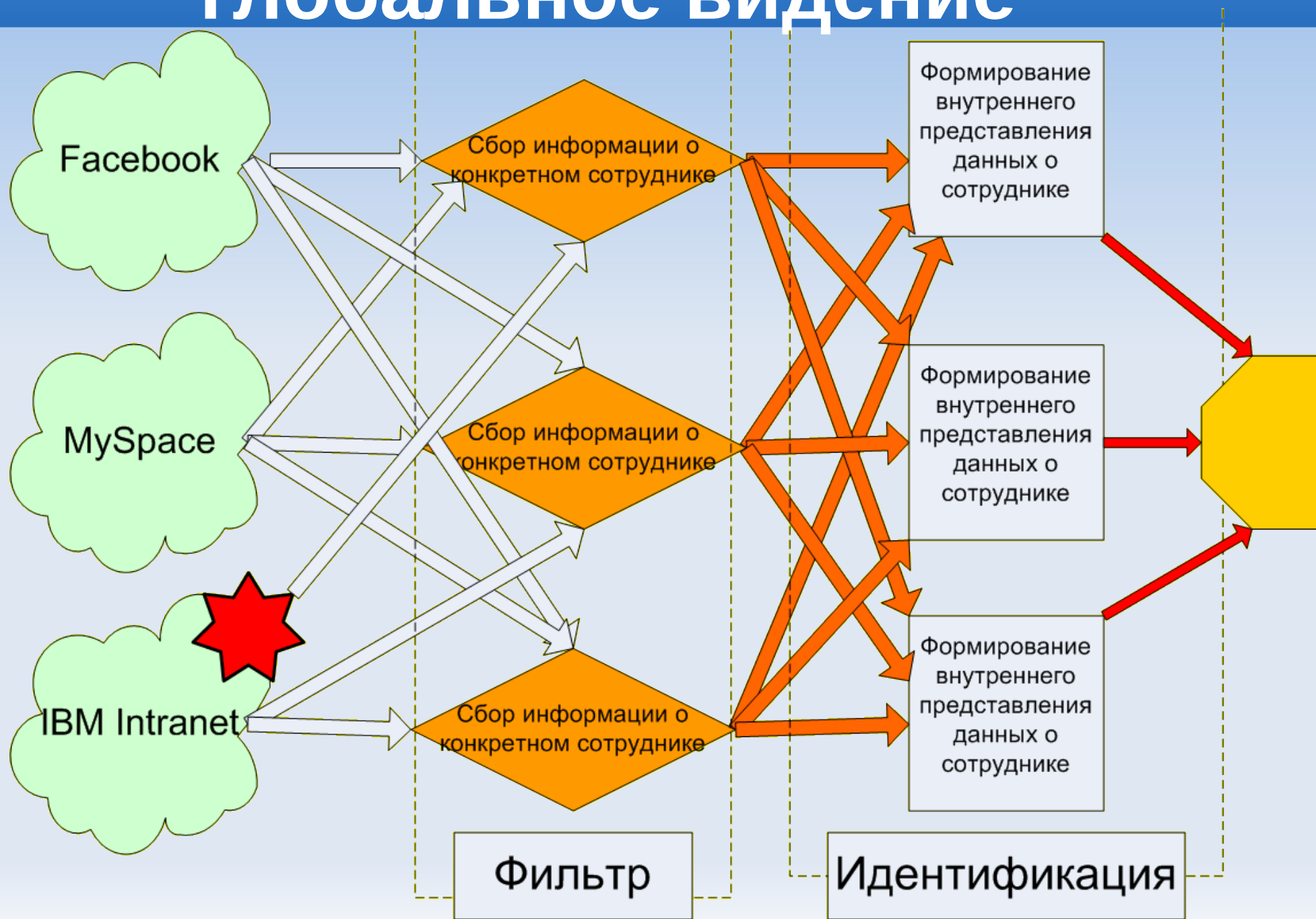
- Изменение политики стоит проводить в рамках бизнес-политики в целом, и рамках присутствия в интернете и в соц сетях в частности
- Благодаря замыканию, мы можем установить область видимости всех пользователей целевой группы.
- В частности поставить как верхний предел саму группу-представитель фирмы  
(М.Н.Штейнберг : "В конечном итоге это все равно произойдет в ближайшие несколько лет, и этот вариант звучит вполне правдоподобно")



# Алгоритм - "отобрази-и-сверни" (MapReduce)



# Алгоритм глобальное видение



# Алгоритм глобальное видение(2)

- Агрегация сетей-графов в MapReduce является классическим примером использования данного алгоритма, именно благодаря подобному представлению данных в нашей модели
- **Важно(!)** свойство замыкания дает нам возможность выполнить и map, и reduce по всем сетям, внешне не делая между ними различия

(Л.Н.Столяров: "Есть только два типа сетей, это первый – вычислительная сеть, именно она тут и необходима")

# Алгоритм глобальное видение(3)

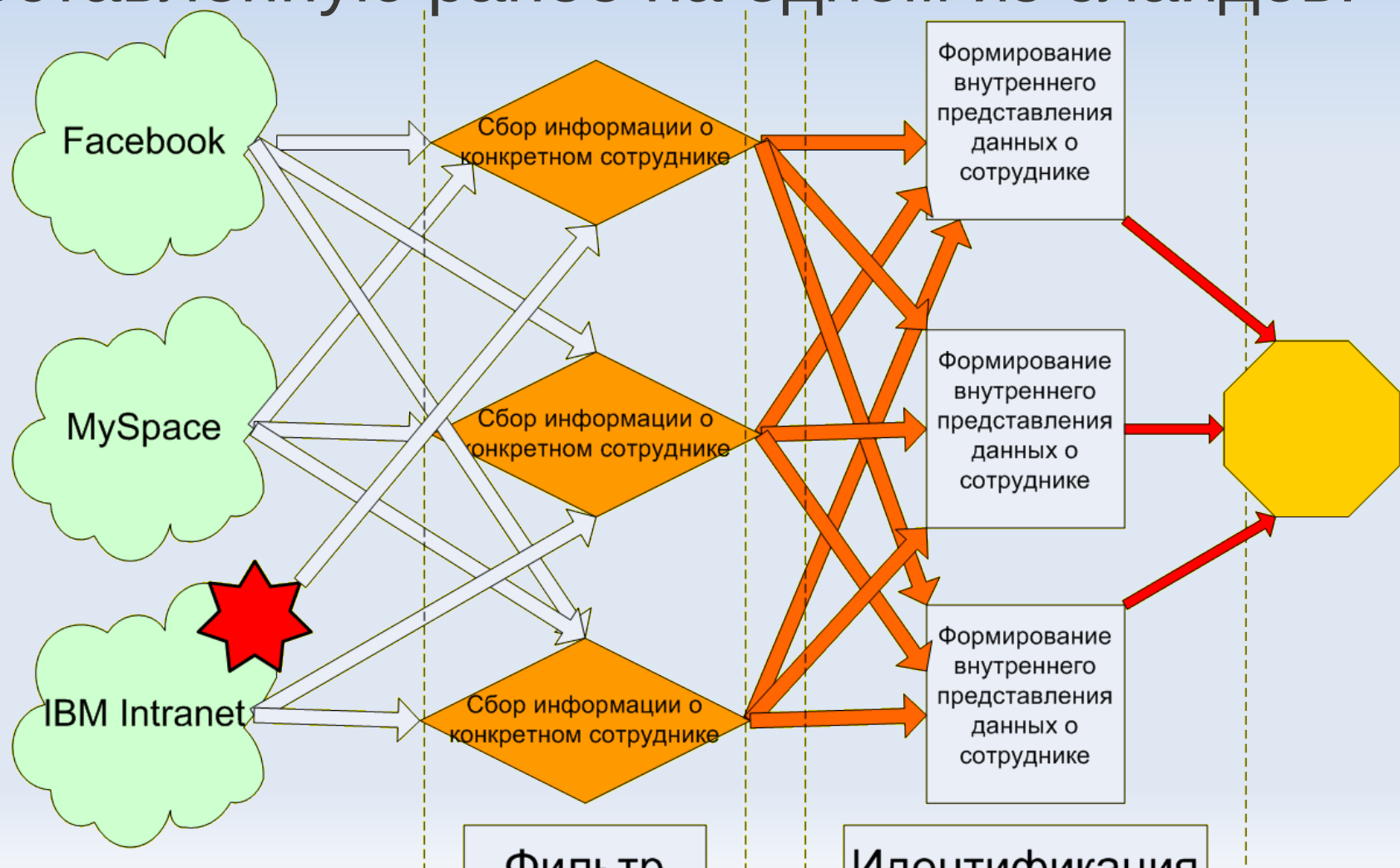
Важнейшие свойства из нашей модели и алгоритма:

- Замыкание
- Параллельность
- Наличие признанного паттерна
- Обеспечение консистентности\* данных

( В.Э Вольфенгаген: "Сама идея обеспечения консистентности данных такой системы безусловна правильная, но я бы оценил ее полное аналитическое решение в объем докторской диссертации")

# Алгоритм глобальное видение(4)

Теперь обратим внимание на звездочку поставленную ранее на одном из слайдов:



# Алгоритм глобальное видение(5)

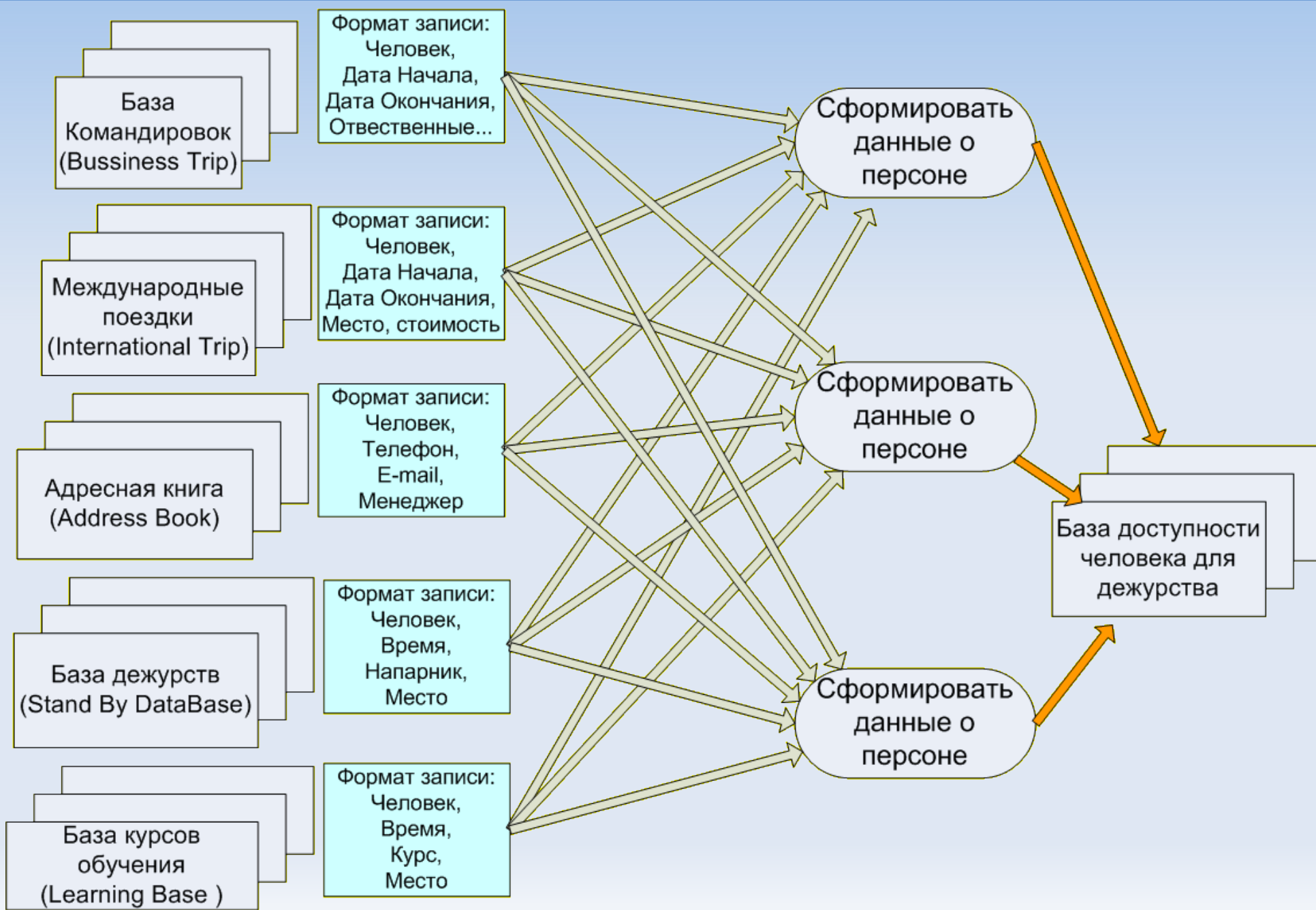
- Эта звездочка показывает слабое место нашего алгоритма и модели:  
мы обращаемся с интранетом так же как и социальной сетью, которая обладает API и социальной структурой, которая позволяет построить семантическую структуру
- Поэтому принципиально разрешить именно эту проблему, показав, что мы можем построить для него Map и Reduce, чтобы добавить его в наш алгоритм и не нарушить свойство замыкания

# Программа\*

- Изменение уровня доступа
- Агрегация данных
- Внутреннее представление
- Демонстрация работы

\*Программа была написана для интранет от IBM Russia System and Technology Laboratory

# Программа(2)





# Программа(3)

Сложность состоит именно в преобразовании данных из внутреннего формата Lotus Domino баз в наш инвариантный формат, поэтому было использовано два различных подхода

Lotus Notes Java API – Notes.jar

ODBC-JDBC мост с NotesSQL IBM Driver

# Программа(4)

- Хотя в целом метод с Lotus Java API показывает лучшие результаты, это скорее следует отнести к недостаточной проработке структуры получаемого реляционного представления

Процент распознавания нашим отображением(Map) доступных баз					
	Командировки	Междн. поездки	Адресная книга	База дежурств	Курсы обучения
ODBC-JDBC Bridge	87	91	94	89	86
Lotus Java API	91	93	95	91	89

# Результат

- Разработан протокол на примере системы IBM Lotus Domino в рамках автоматизации бизнес процессов (в IBM Russia System and Technology Laboratory )
- Организовано два варианта кроссплатформенной репрезентации и передачи данных: XML(для моста), nsf (для Lotus API).
- Гипотеза о представлении не противоречит статистическим данным в пределах ошибки ( 10% для моста и 8% для Lotus API)

# Анализ. Дальнейшее направление исследований.

- Анализ бизнес-процессов при проектировании системы разграничения прав и доступа.
- Обеспечение консистентности данных.
- Необходимо разработать общий распределенный паттерн для интеграции обладающий свойством замыкания