Распределенная файловая система

Почему Hadoop ?

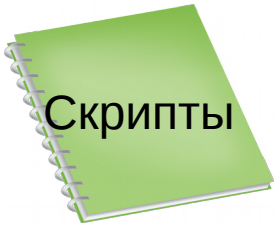
- Практически эталонная реализация концепции Big Data
- Сотни и тысячи пользователей.
- Open source
- Написана на java

Hadoop предоставляет все базовые компоненты Big Data

- Распределенная файловая система – HDFS
- BigTable – HBASE
- Map Reduce – Hadoop map reduce

Стек технологий Hadoop

Интерфейс



Консоль

Планировщики

Map reduce

PIG

Storage

HDFS

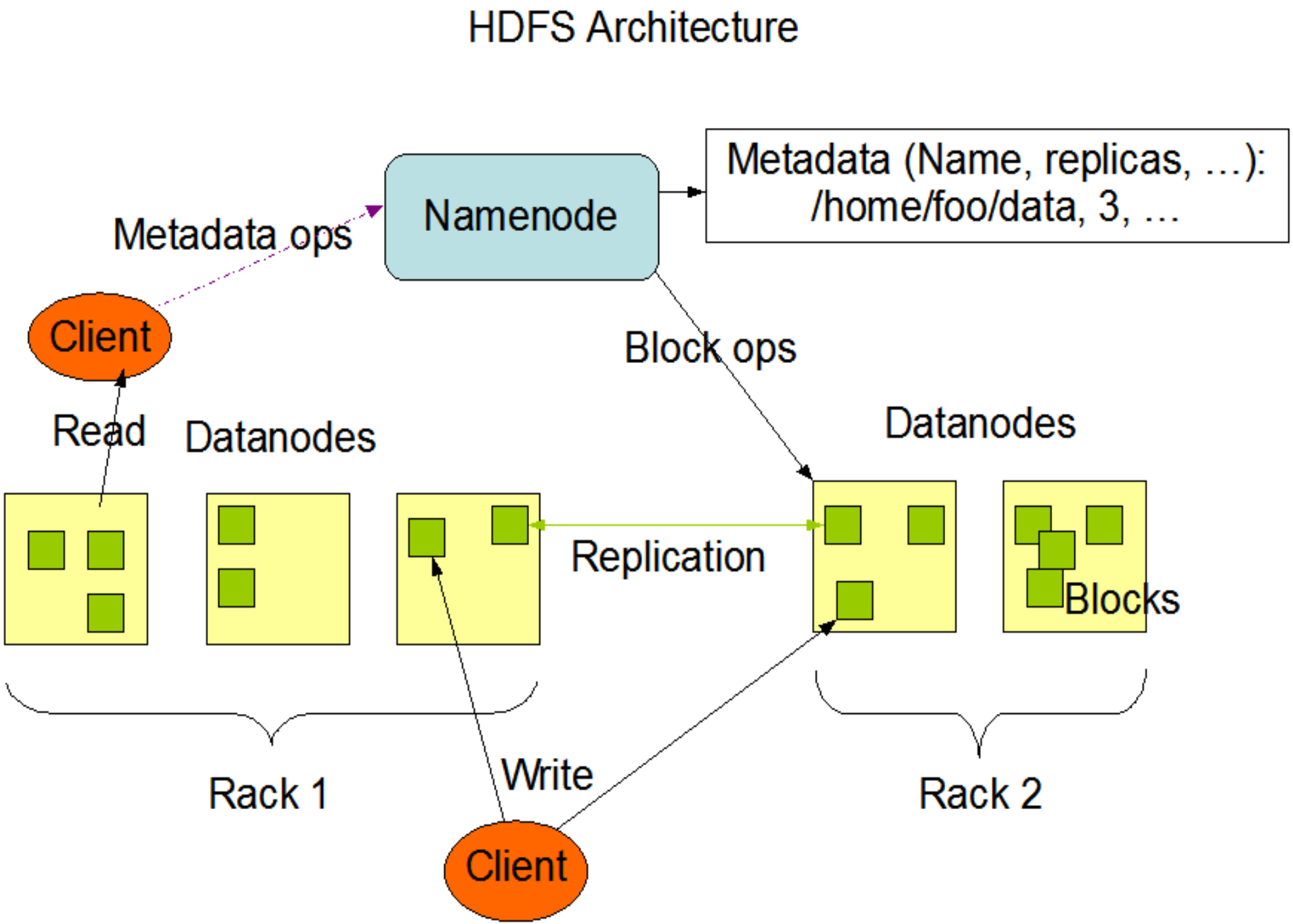
- Реализация распределенной файловой системы
- Позволяет хранить ОЧЕНЬ большие файлы
- Предназначена для параллельной поточной обработки файлов
- Работает на недорогом железе (отказоустойчивость основана на репликации средствами FS)

Ситуации когда HDFS не надо применять

- Требуется малое время доступа к произвольным данным
- Требуется FS для хранения большого количества мелких файлов
- Много writers, которые пишут в файл одновременно
- Требуется менять произвольные участки файла (запись со смещением)

Архитектура HDFS

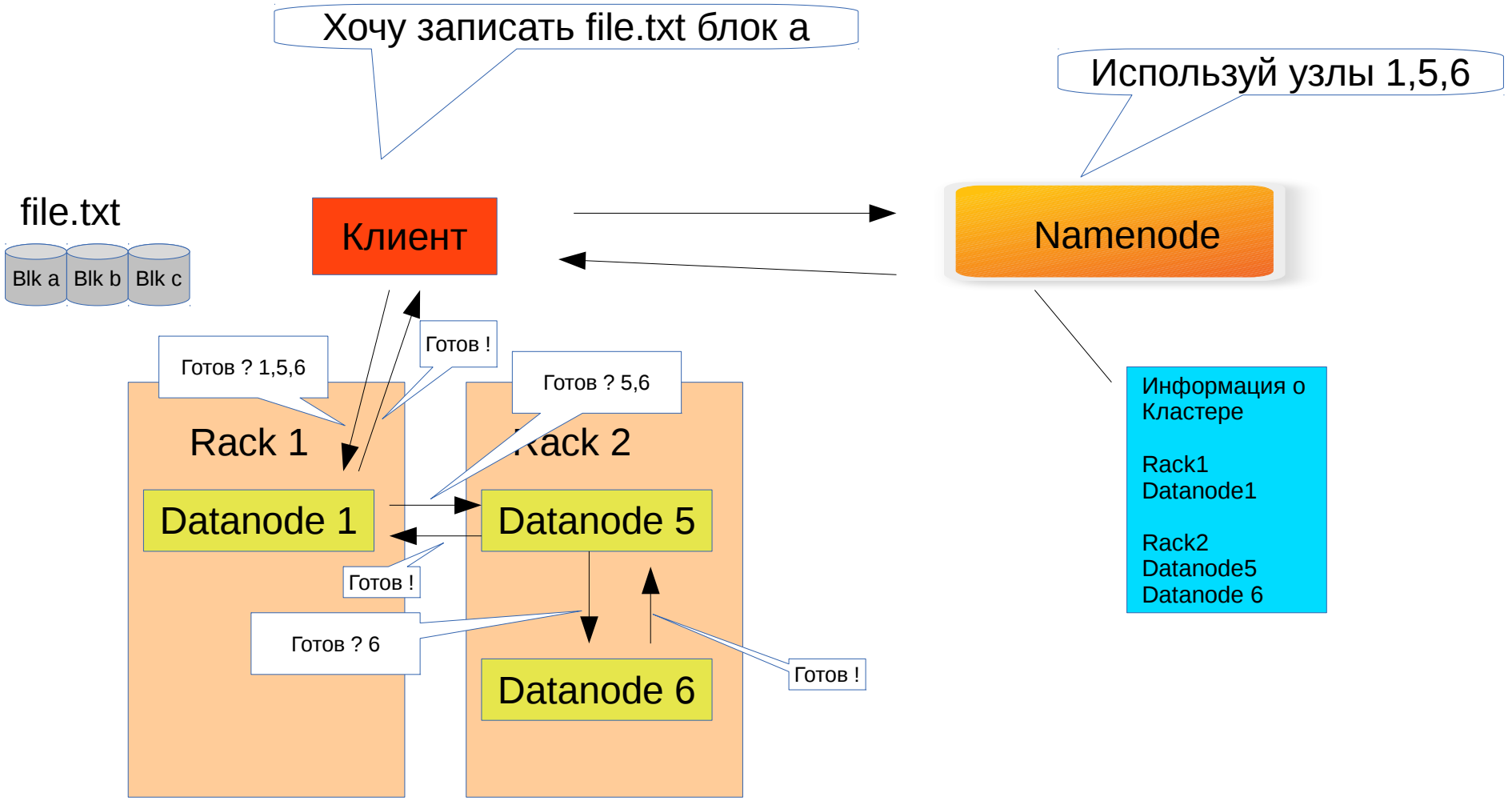
- Кластер HDFS состоит из узлов двух типов:
- Datanodes – хранят файлы
- Datanodes могут быть сгруппированы в Racks (группы серверов кластера расположенные недалеко друг от друга)
- Namenodes – хранят метаданные о файлах
- В каждый момент времени активен один узел Namenode


HDFS Architecture
Namenode
Metadata (Name, replicas, …):
/home/foo/data, 3, …
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client

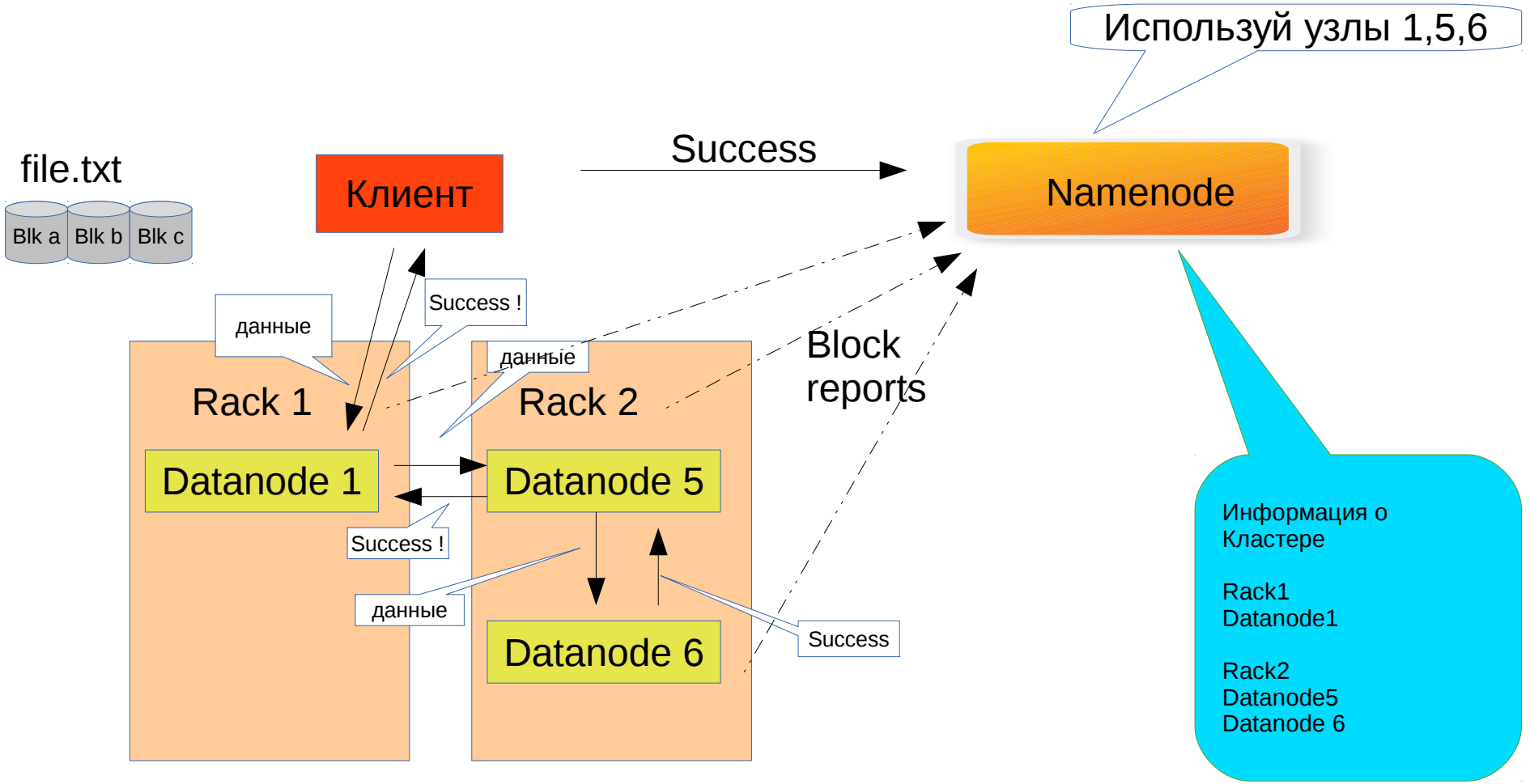
Файл в HDFS

- Файл разбивается на большие блоки (по умолчанию 64 МБ)
- Каждый блок хранится на другом узле кластера HDFS
- В зависимости от настроек репликации для каждого блока хранится определенное количество реплик на других узлах
- Namenode хранит информацию обо всех блоках каждого файла

Подготовка к записи в файл



Потоковая запись в файл



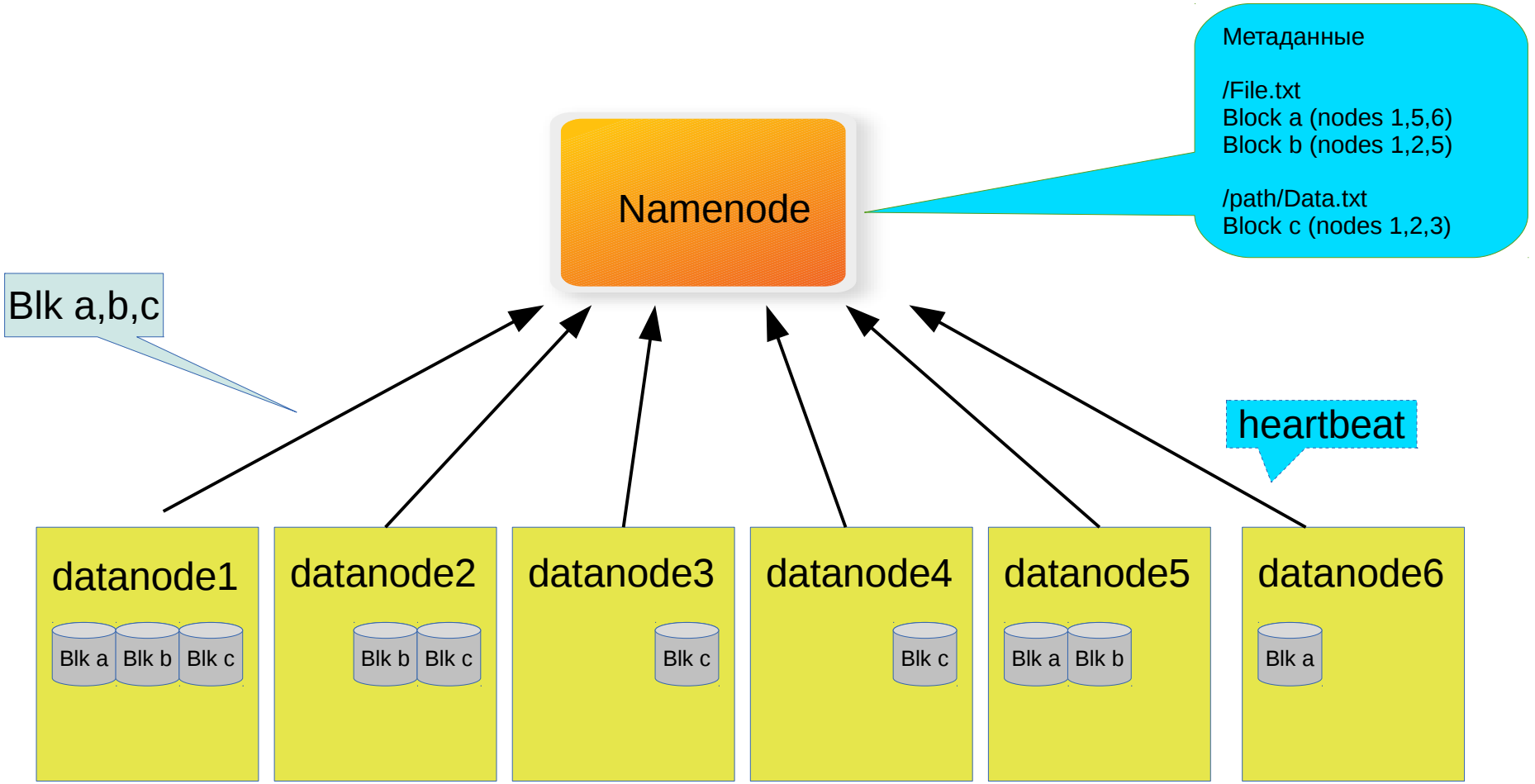
Security в HDFS

- Используется POSIX модель
- Каждый файл имеет владельца и группу
- Права задаются тремя группами – для владельца, для группы и для всех остальных
- Используется для разграничения доступа в “дружественном” окружении.
- Аутентификация при использовании обычного клиента не производится.

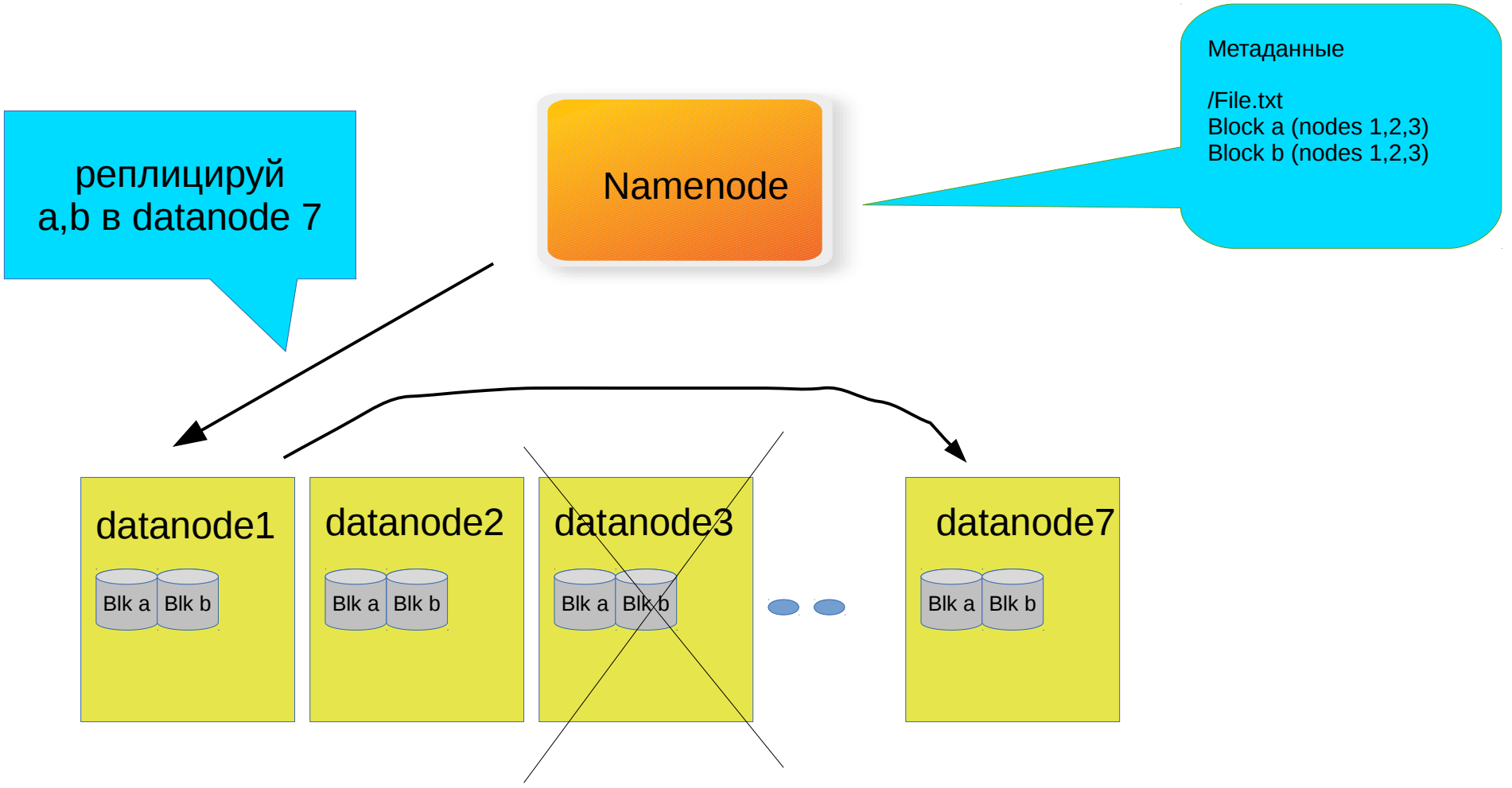
Отказоустойчивость datanode

- Если в hdfs есть разбиение на racks то по умолчанию каждый блок хранит копии на двух разных racks
- Каждый datanode постоянно (раз в 3 секунды) посылает heartbeat сигнал NameNode-у
- Каждый 10-й сигнал содержит информацию о блоках данных размещенных на datanode
- В случае если сигнал не пришел – Namenode дополнительно копирует данные с упавшего datanode на живые

Взаимодействие namenode и datanode



Реплицирование потерянной информации



Отказоустойчивость namenode

- Список файлов с правами и разбиением на блоки хранится в памяти
- Каждая операция сначала записывается в файл edit log на диске, потом в память
- Чтобы файл edit log не рос выше определенного предела – используется secondary name node
- Secondary namenode периодически забирает с namenode файл edit log а также старый дамп списка файлов, применяет edit log к дампу и отсылает назад актуальный дамп памяти.
- В случае падения namenode – администратор запускает новый экземпляр, который берет дамп памяти и edit log
- После того как новый namenode получит все отчеты о блоках от datanode а также сформирует новую схему файлов в памяти – он готов к работе

Доступ к файлам HDFS – командная строка

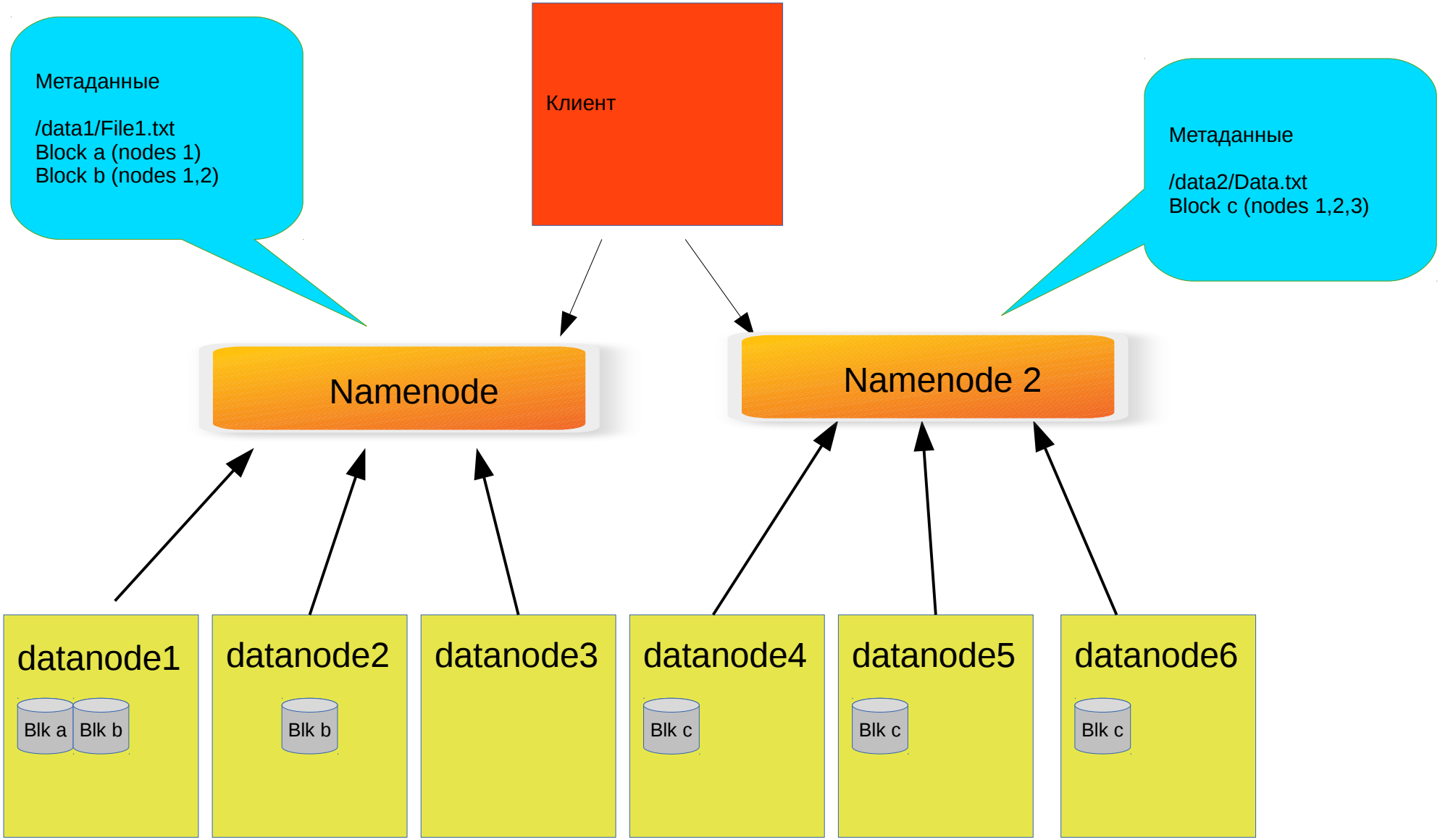
- hadoop fs -copyFromLocal input/docs/data.txt
- hadoop fs -copyToLocal input/docs/data.txt
- hadoop fs -mkdir books
- hadoop fs -ls .

Доступ к файлам – java api

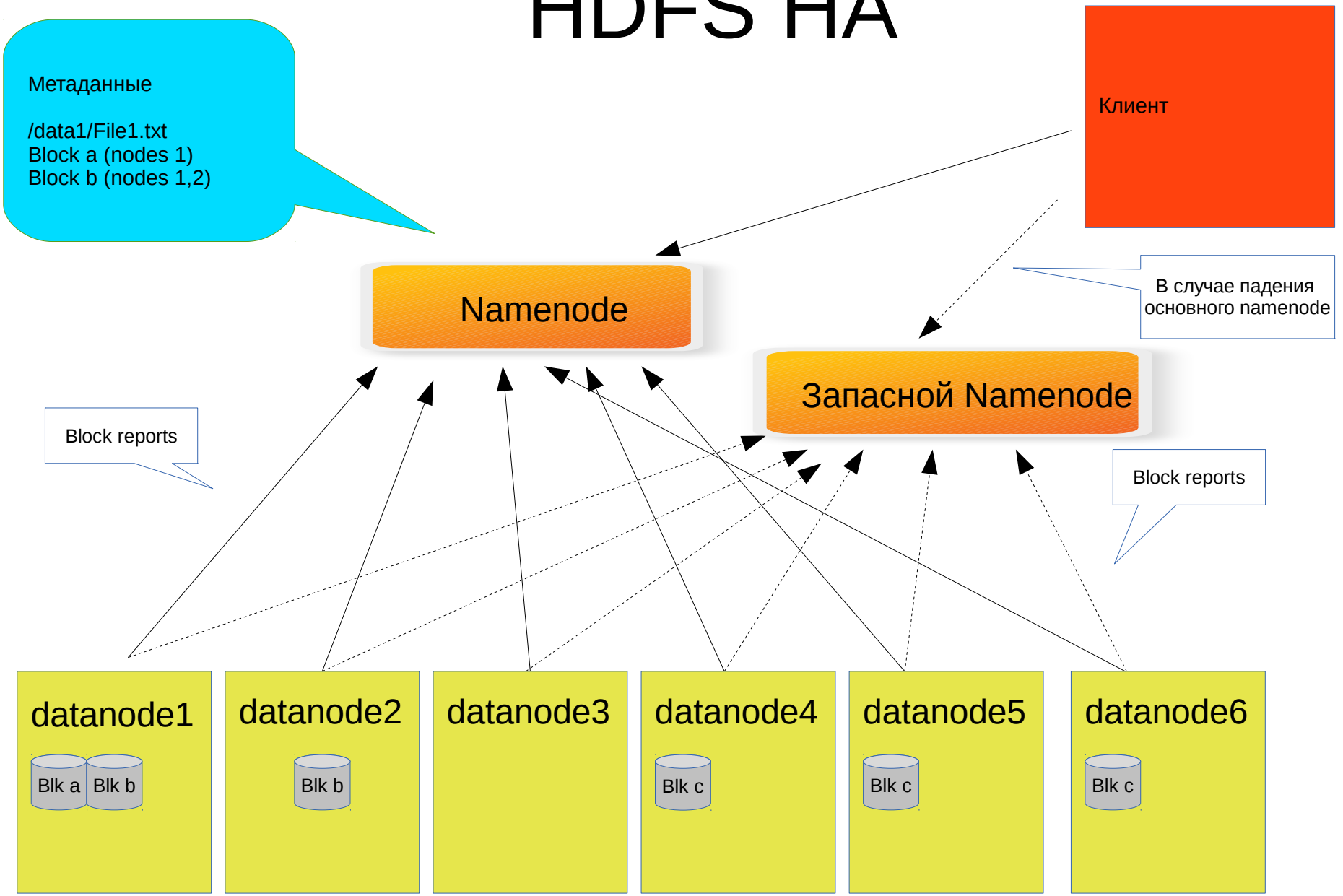
- public class FileSystemCat {

```
    public static void main(String[] args) throws Exception {
        String uri = args[0];
        Configuration conf = new Configuration();
        FileSystem fs = FileSystem.get(URI.create(uri), conf);
        InputStream in = null;
        try {
            in = fs.open(new Path(uri));
            IOUtils.copyBytes(in, System.out, 4096, false);
        } finally {
            IOUtils.closeStream(in);
        }
    }
}
```

HDFS Federation



HDFS HA


Метаданные
/data1/File1.txt
Block a (nodes 1)
Block b (nodes 1,2)
Клиент
Namenode
В случае падения основного namenode
Запасной Namenode
Block reports
Block reports
datanode1
Blk a
Blk b
datanode2
Blk b
datanode3
datanode4
Blk c
datanode5
Blk c
datanode6
Blk c

Quorum Journal Manager