

Распределенные информационные системы

Технология DDS – 4

Вопросы

- «Более хитрые» QoS в DDS:
 - Состояние компонентов (liveliness)
 - Состояние экземпляров (deadline)
 - Максимальное время доставки (latency budget)
 - Приоритет (priority)
 - Владение (ownership, ownership strength)
 - Фильтрация по времени (Time-based filtering)
 - Фильтрация по значению (Content-Based filtering)
- «Лучшие практики»

Liveliness

QOS

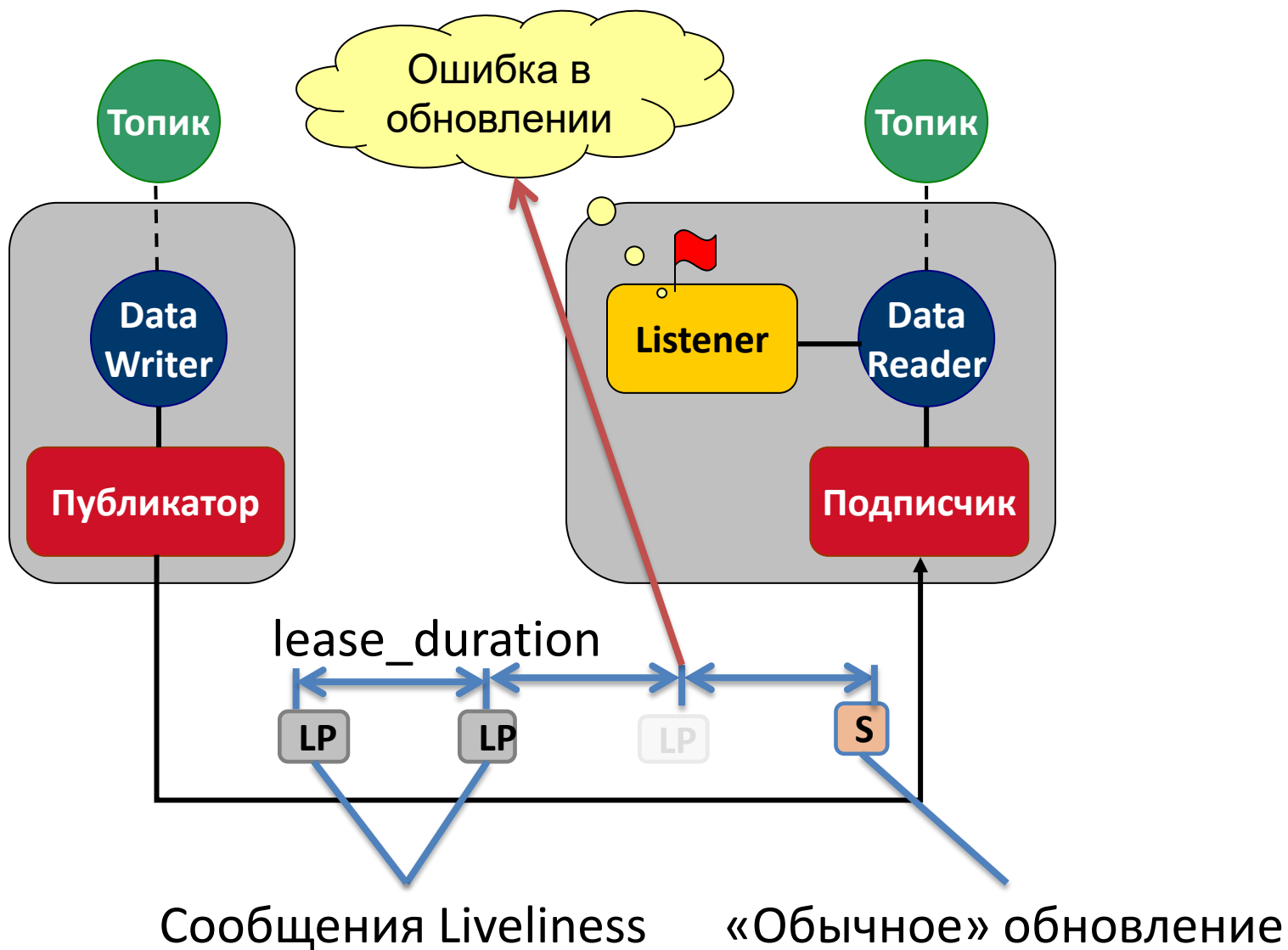
Состояние компонентов DDS – 1/6

- QoS отслеживания состояния компонентов системы (Liveliness QoS) позволяет отслеживать существование, статус и активность компонентов DDS (для Participant, Reader, Writer)
- Упрощенно можно сказать, что отвечает на вопрос «является ли отсутствие новостей хорошей новостью?»
- Для определения состояния используются специальные Liveliness Packets (аналог heartbeats)

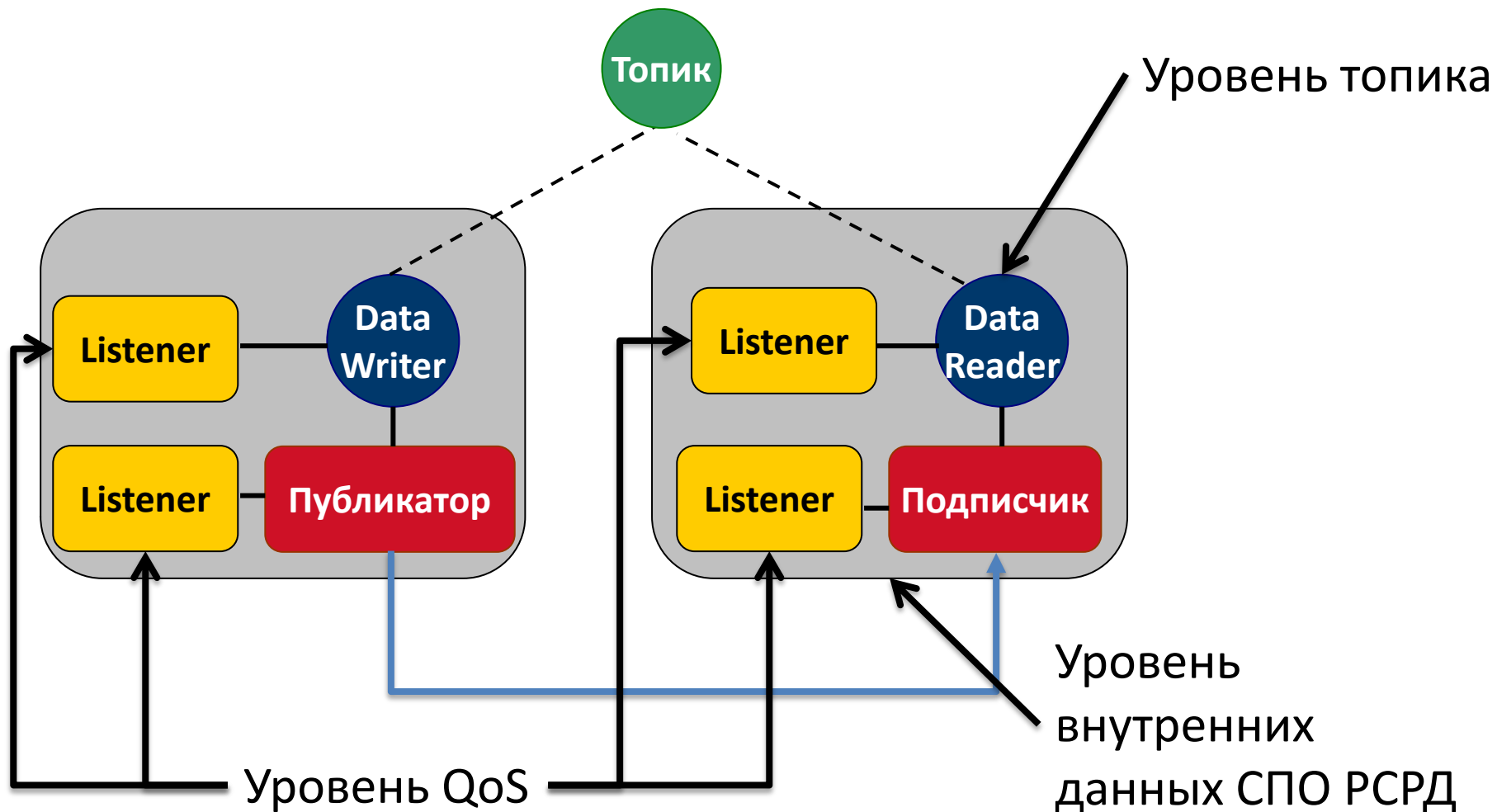
Состояние компонентов DDS– 2/6

- Возможные уровни использования, определяют, кто отвечает за генерацию пакетов
 - AUTOMATIC
 - Пакеты генерируются инфраструктурой автоматически
 - MANUAL_BY_PARTICIPANT
 - MANUAL_BY_TOPIC
 - Генерация пакетов переложена на приложения

Состояние компонентов DDS – 3/6



Состояние компонентов DDS – 3/6



Состояние компонентов DDS – 4/6

- QoS используется совместно с исключительным владением экземплярами:
 - Writer считается владельцем только если он «ЖИВ»
 - Как только DDS обнаруживает факт пропуска прихода сообщений liveliness, она «отбирает» у писателя все его экземпляры и назначает владельцем следующего (с учетом их приоритетности)

Состояние компонентов DDS – 5/6

- Не следует использовать термин `deadline`, чтобы не спутать с `QoS Deadline`
- Периоды посылки сообщений `liveliness` конфигурируемы
- При использовании уровня `MANUAL_*` приложение должно посылать сообщения `liveliness`:
 - В явном виде с помощью вызова метода `assert_liveliness`
 - В неявном виде с помощью посылки семпла

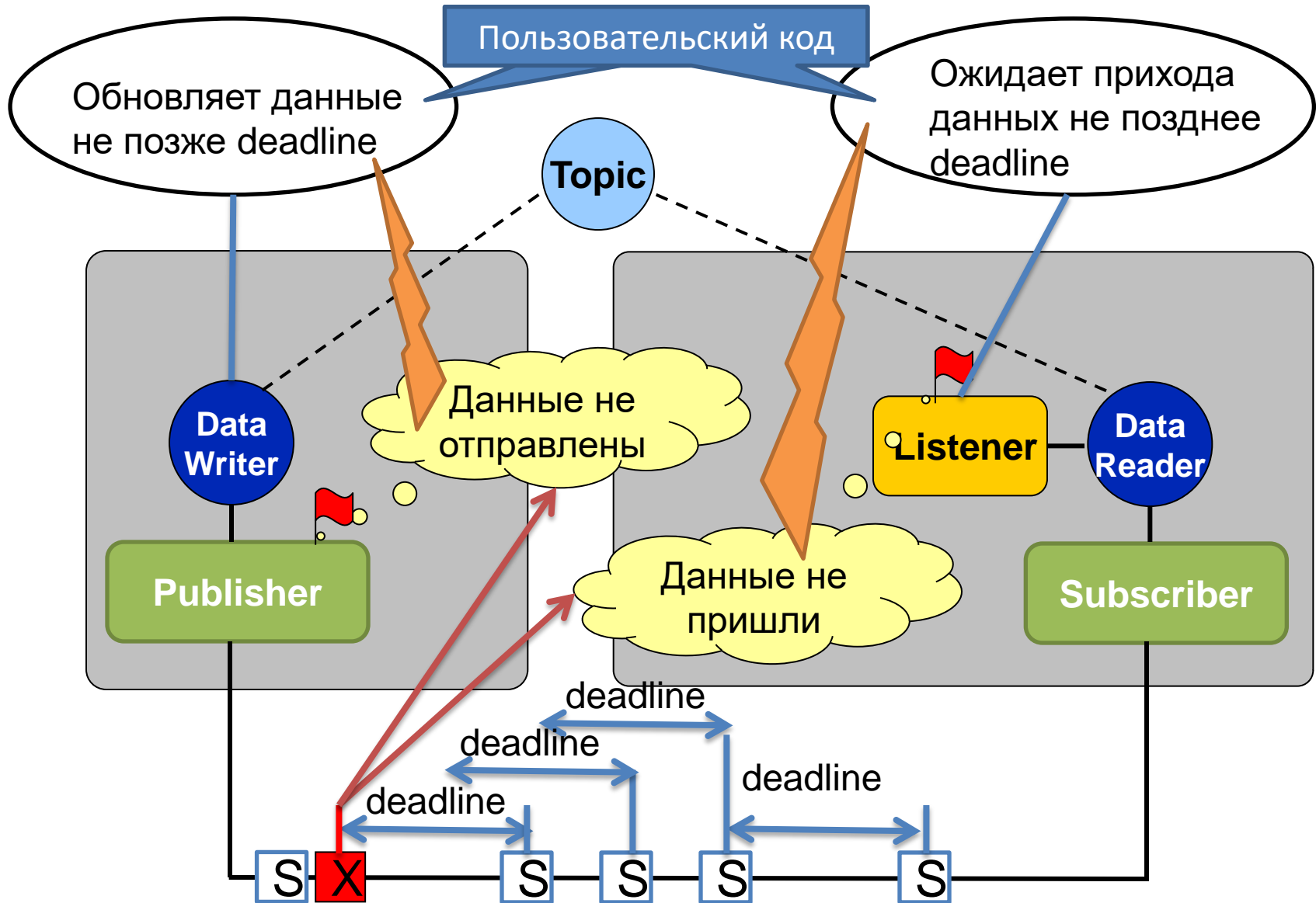
Состояние компонентов DDS – 6/6

- MANUAL_BY_PARTICIPANT определяет состояние на уровне DomainParticipant
- MANUAL_BY_TOPIC определяет состояние на уровне DataWriter
- Соглашение о настройках QoS происходит с учетом предлагаемых:
 - Уровня (можно сдвинуть выше):
 - AUTOMATIC < MANUAL_BY_PARTICIPANT < MANUAL_BY_TOPIC
 - Времени задержки (можно сделать меньше):
 - Предложенное время < установленного времени

Состояние экземпляров – 1/2

- QoS отслеживания состояния экземпляров (Deadline QoS) позволяет отслеживать статус и активность экземпляров
- Если данные по экземпляру не обновляются как установлено, то приложение получает сигнал об этом

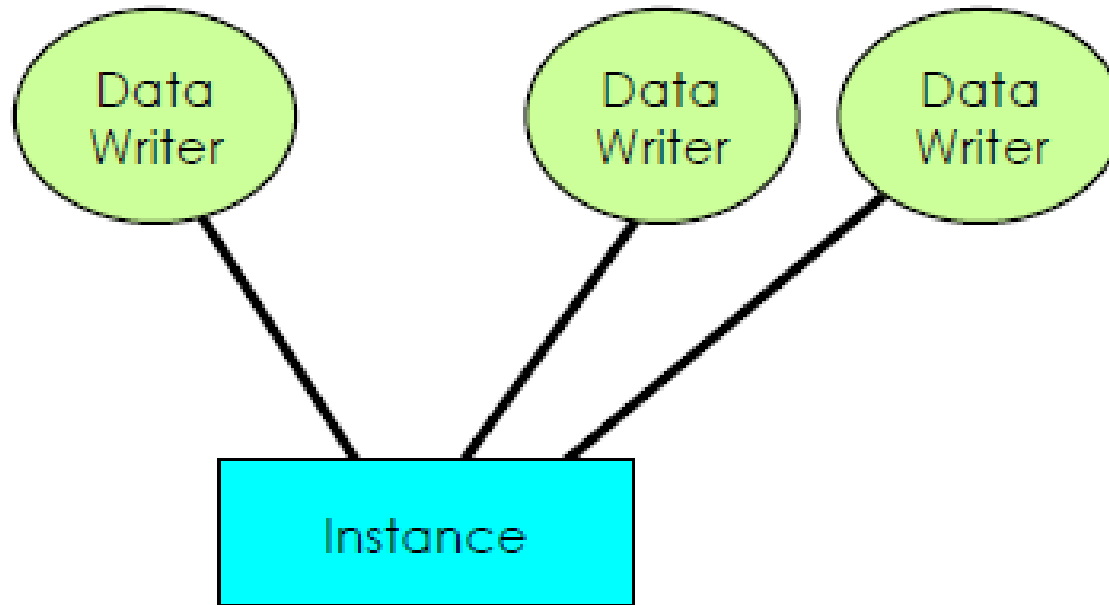
Состояние экземпляров – 2/2



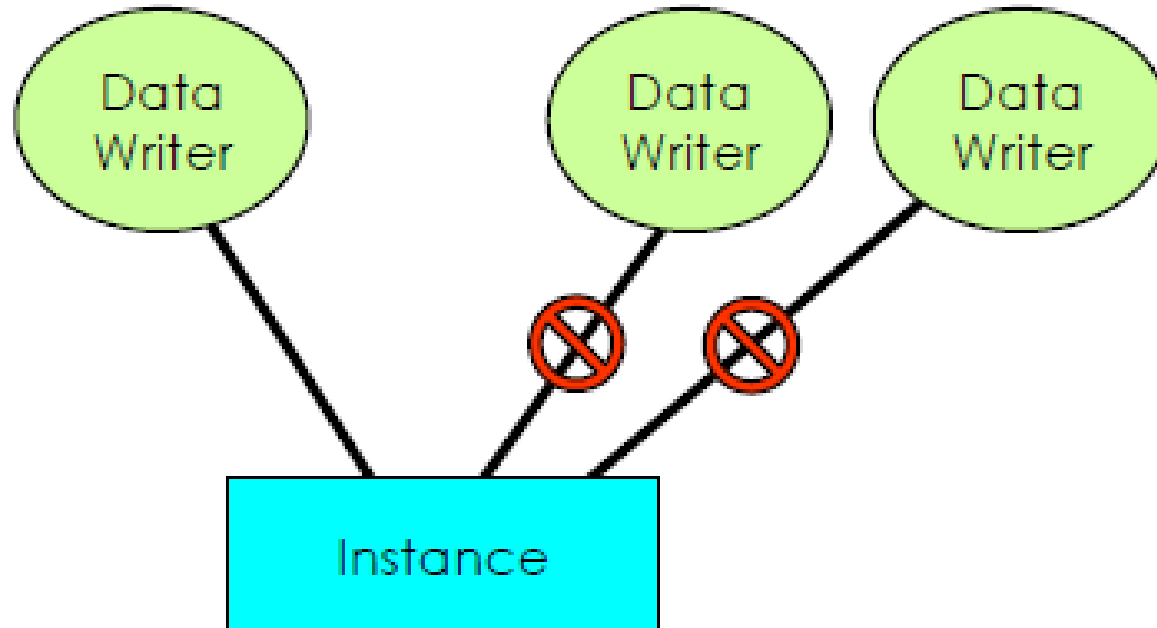
Владение – 1/6

- Владение (Ownership) устанавливается на уровне экземпляра (или топика, если для типа данных не установлен ключ). Возможно установить следующие варианты владения:
 - SHARED: Совместное владение, когда любой участник, опубликовавший топик, может обновлять экземпляр. По действию такой режим эквивалентен отключению использования данной QoS
 - EXCLUSIVE: Исключительное владение, когда только один источник (writer) может обновлять данные по экземпляру

Владение: Shared – 2/6



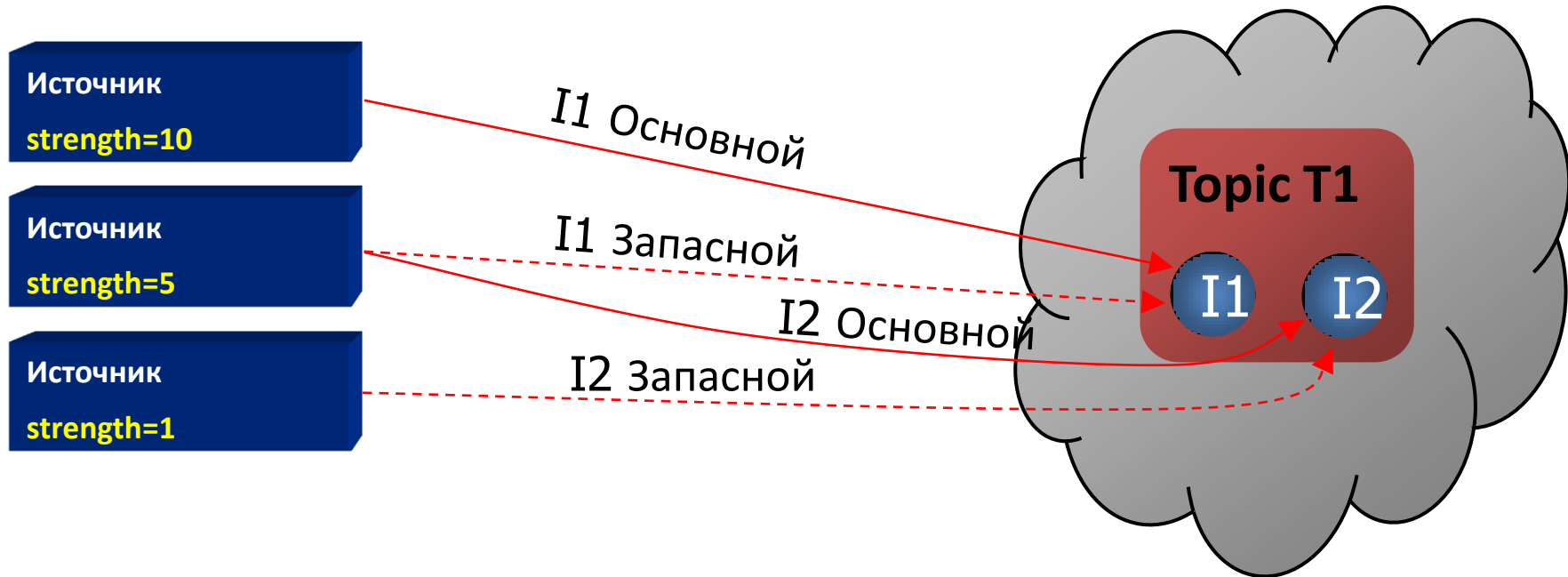
Владение: Exclusive – 3/6



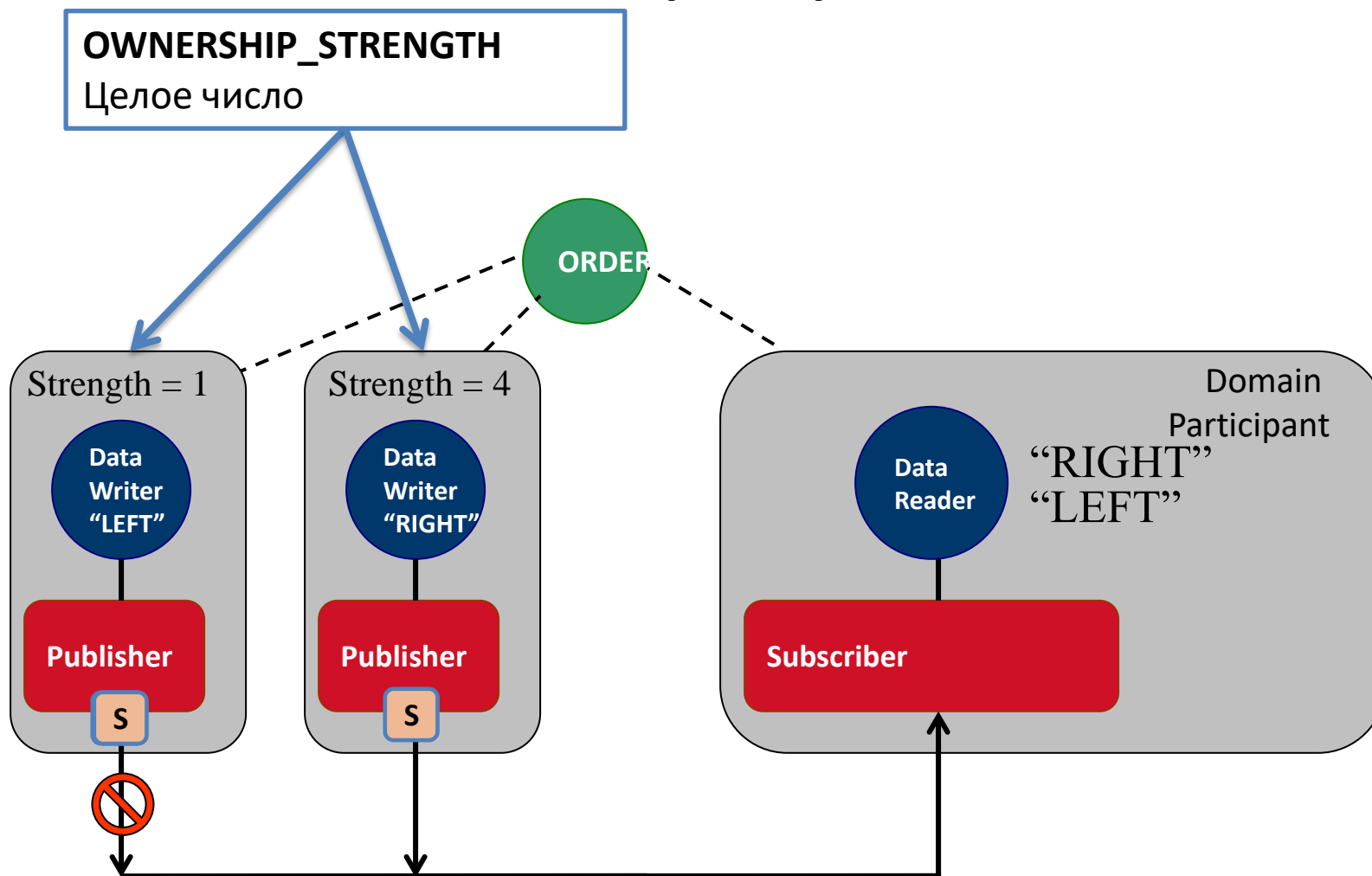
Владение – 4/6

- Приоритет владения (Ownership Strength) устанавливается для источника (writer) и применяется к его экземплярам (или топикам, если для типа данных не установлен ключ)
- Только один источник (writer), имеющий максимальный приоритет, может обновлять данные по экземпляру
- Владение автоматически передается:
 - При появлении более приоритетного источника,
 - Или следующему по приоритету, если:
 - Источник «умер» (согласно QoS liveliness)
 - Пропустил отправку данных (согласно QoS Deadline)
 - Перестал публиковать топик или экземпляр

Владение: Приоритет – 5/6



Владение: Приоритет – 6/6

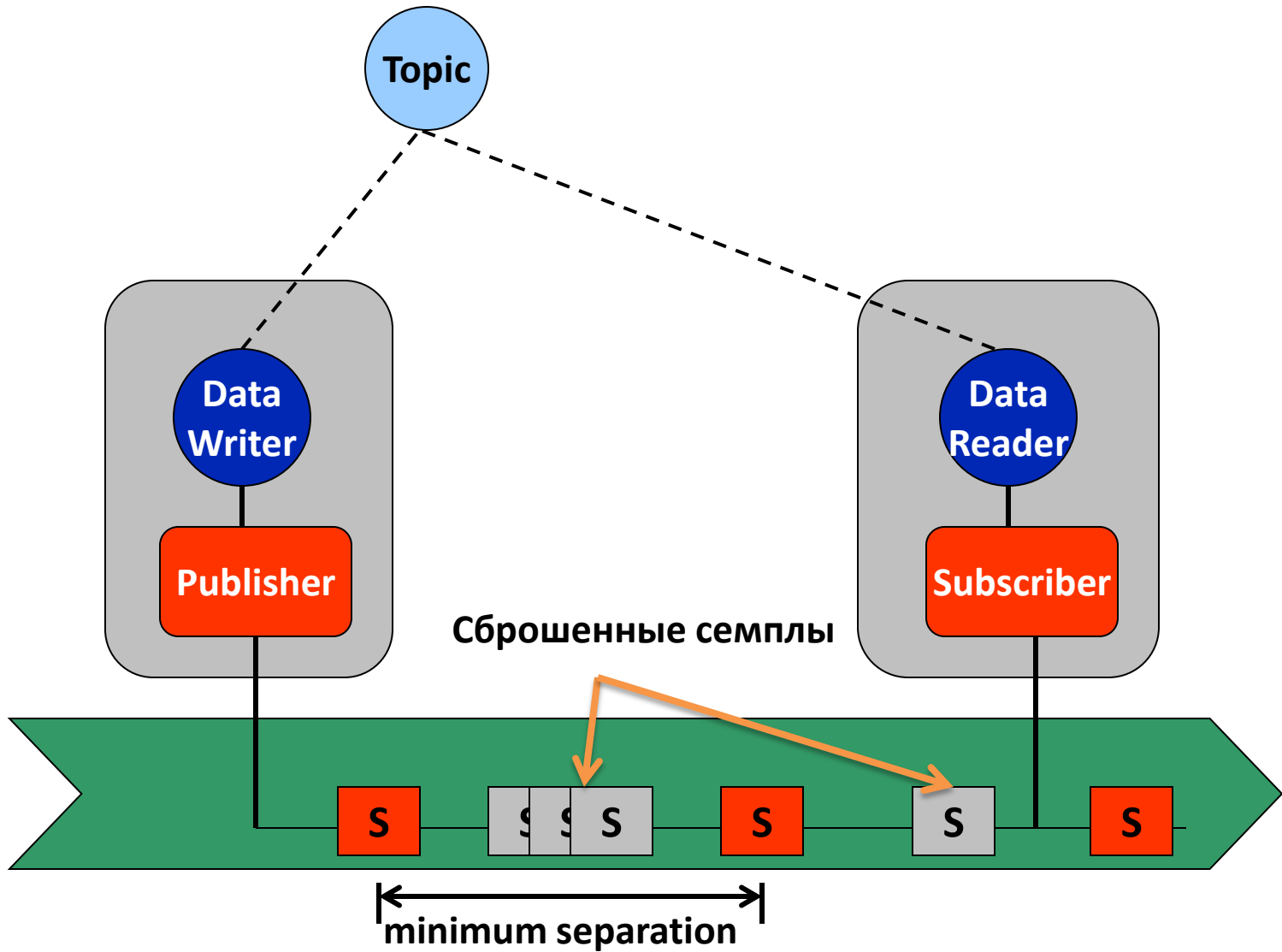


Примечание: Только для Ownership = Exclusive

Фильтрация по времени – 1/2

- Фильтрация по времени (Time_based_filter) позволяет подписчикам указывать, что они не желают получать данные слишком часто. Это позволяет сократить количество получаемых семплов и сэкономить время на их обработку
- DataReader указывает, что:
 - Ему не обязательно получать все семплы
 - Однако он хочет получать не меньше одного семпла в каждый период (minimum_separation)

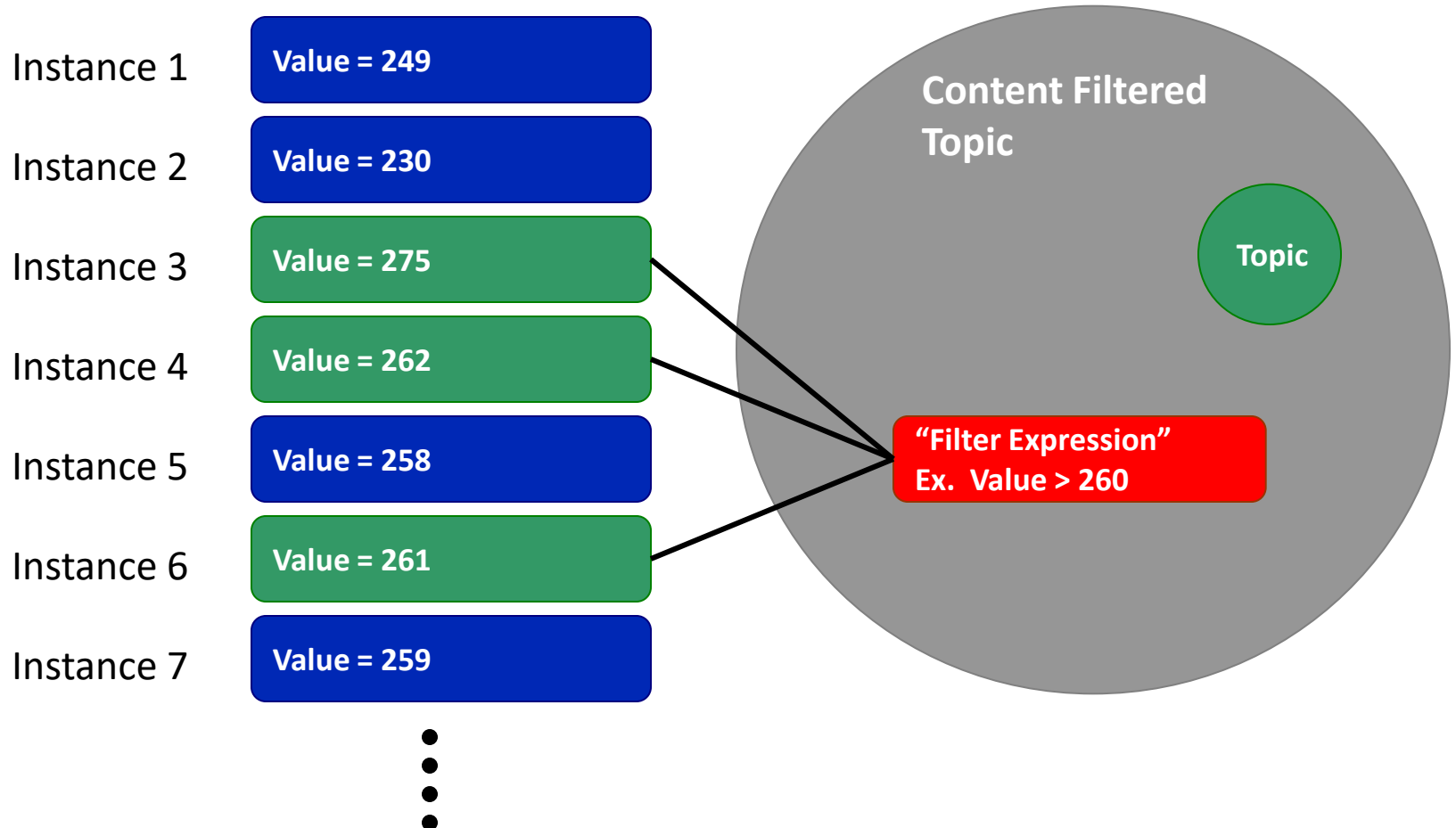
Фильтрация по времени – 2/2



Фильтрация по значению – 1/2

- Фильтрация по значению (Content-based filtering) позволяет подписчикам указывать, семплы с какими значениями полей они хотят получать. Это позволяет:
 - Исключить из видимости некоторые экземпляры
 - Сократить количество получаемых семплов и сэкономить время на их обработку
 - Получать только «особенные» семплы (индикация событий)

Фильтрация по значению – 2/2



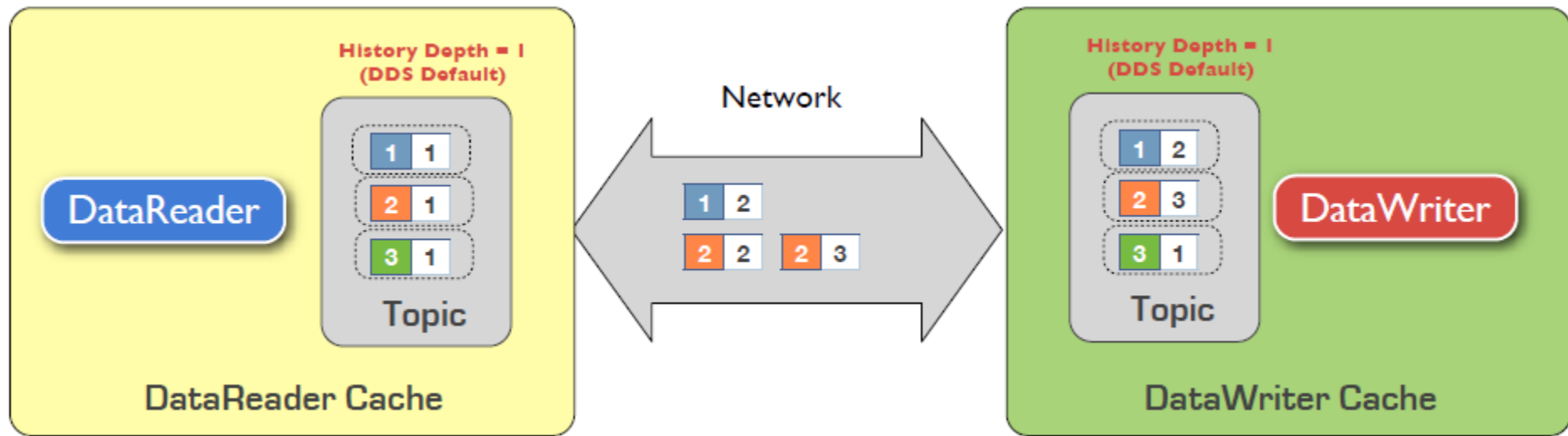
Хранение истории – 1/4

- Хранение истории (history QoS) назначает количество семплов экземпляра, которые хранятся в middleware для DataReader.

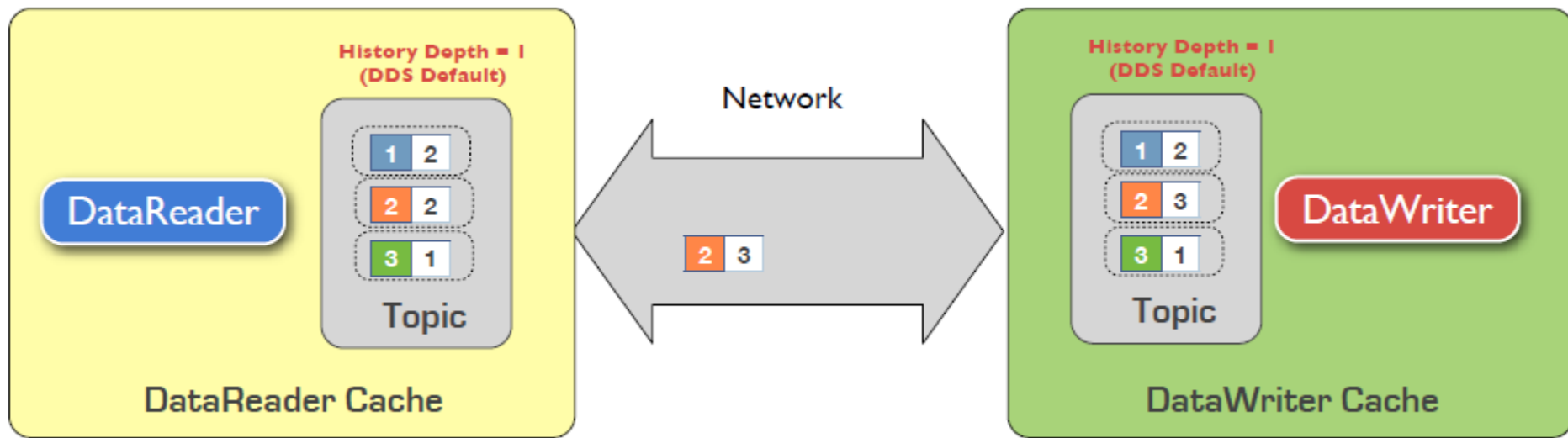
Возможные значения:

- Хранить последние K – в этом случае хранятся последние K семплов. По умолчанию $K = 1$
- Хранить все – в этом случае сохраняются все семплы, отправленные DataWriter, до тех пор, пока:
 - Они не «взяты» (taken) или
 - Не достигнуто ограничение на использование ресурсов

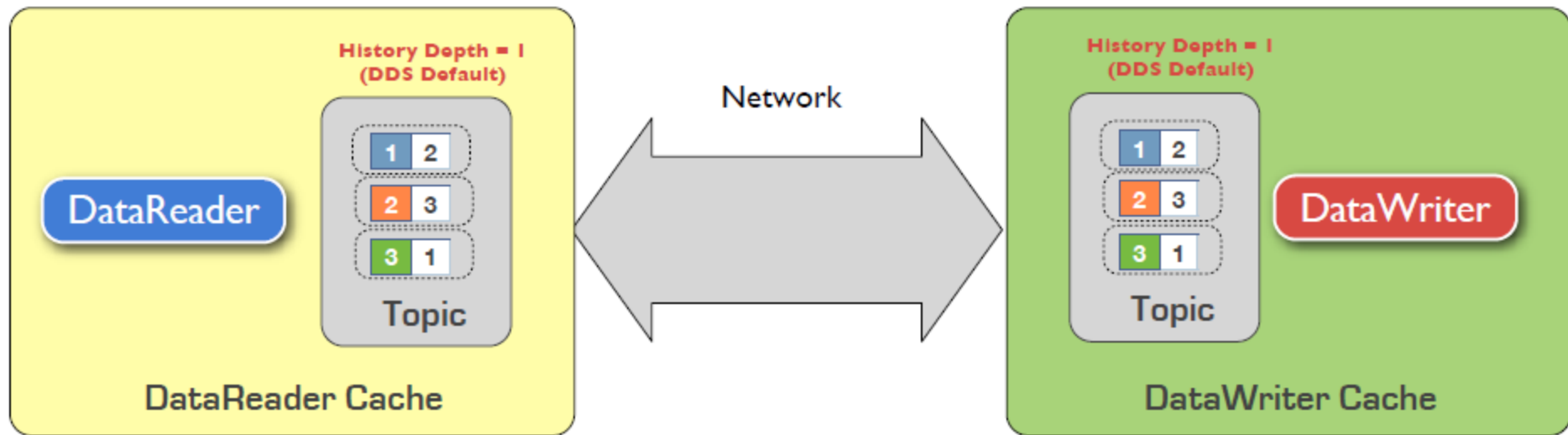
Хранение истории – 2/4



Хранение истории – 3/4



Хранение истории – 4/4



DDS: Durability QoS – 1/2

- В DDS имеется QoS описывающий варианты хранения истории изменений:
 - Volatile (непостоянный, изменчивый)
 - История не хранится
 - Transient_local
 - История хранится локально в памяти отправителя
 - Transient
 - История хранится в сервисе TRANSIENT (в оперативной памяти)
 - Persistent
 - История хранится в постоянной памяти

DDS: Durability QoS – 2/2

PERSISTENT > TRANSIENT > TRANSIENT_LOCAL > VOLATILE

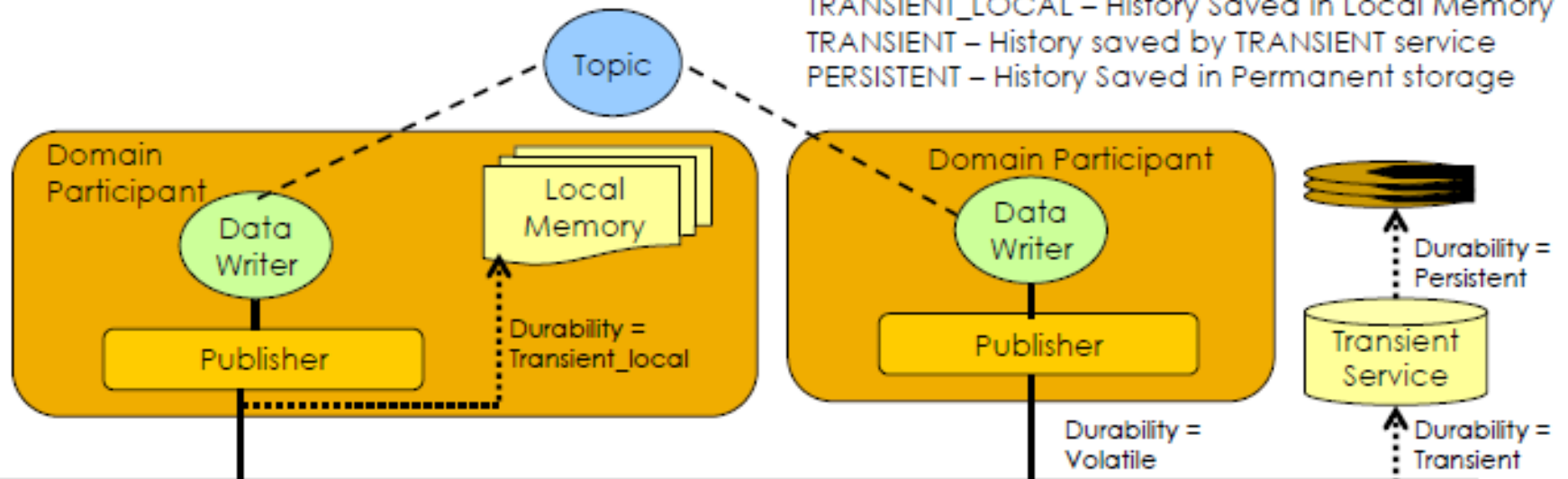
Durability Kind:

VOLATILE – No Instance History Saved

TRANSIENT_LOCAL – History Saved in Local Memory

TRANSIENT – History saved by TRANSIENT service

PERSISTENT – History Saved in Permanent storage



saved in Transient affected by QoS: History and QoS: Resource_Limits

Ограничение использования ресурсов

QOS

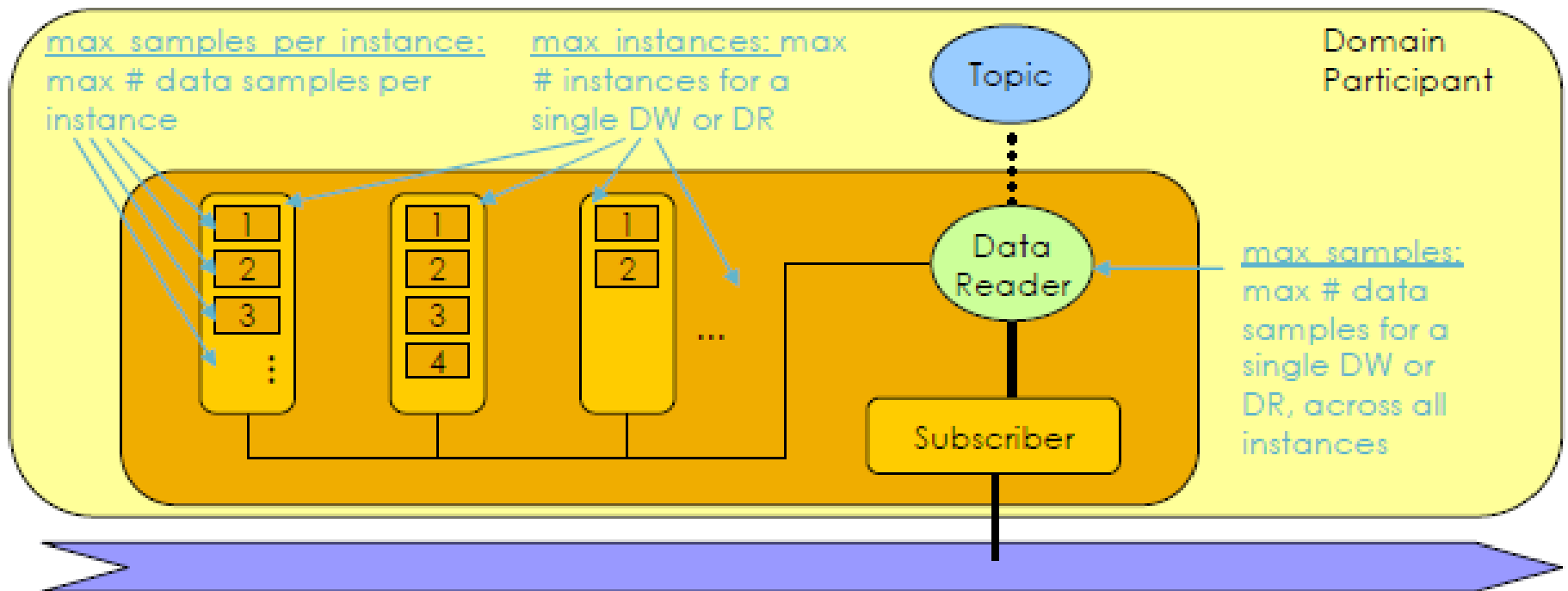
Ограничение использования ресурсов – 1/2

- QoS Ограничение использования ресурсов (ResourceLimitsQosPolicy QoS) позволяет ограничить использование ресурсов приложениями и/или библиотеками DDS
- Ограничение может устанавливаться на уровне:
 - Используемая оперативная память
 - Количество экземпляров
 - Количество семплов:
 - В экземпляре
 - В DataWriter, DataReader

Ограничение использования ресурсов – 2/2

RESOURCE LIMITS

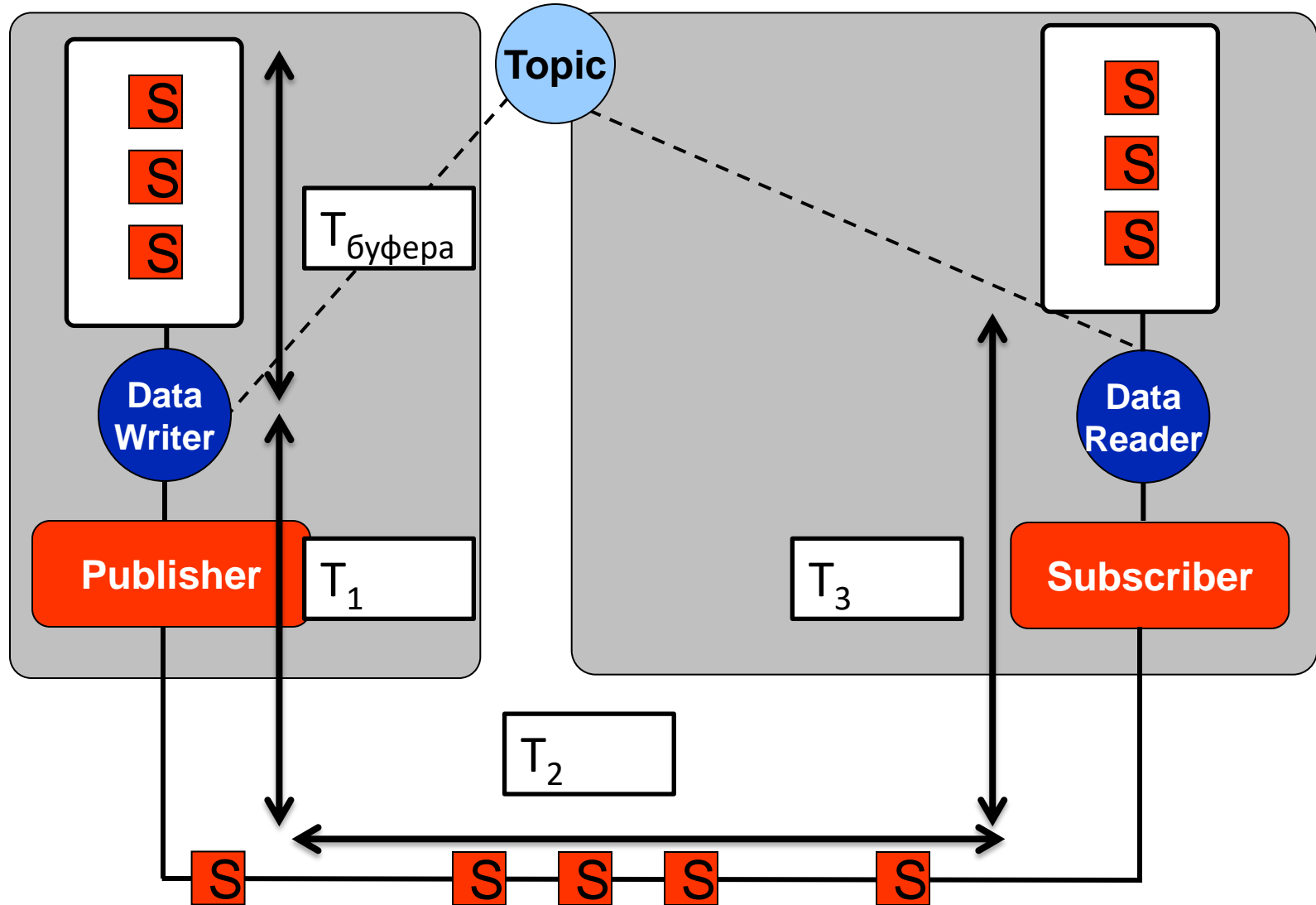
- Specifies the resources that the Service can consume to meet requested QoS.
- The default value is unlimited.



Время на доставку – 1/2

- Максимальное время доставки (Latency_budget) определяет максимально допустимое время задержки данных между их отправкой (операция write) и получением (записью в очередь чтения подписчика)
- Значение по умолчанию – нулевое, что означает, что время доставки должно быть минимизировано
- Данная QoS является «подсказкой» для DDS. Время доставки не мониторится и не может быть явно установлено

Время на доставку – 2/2



Приоритет

- Приоритет (Transport_priority) является подсказкой транспорту, устанавливающей «преимущества» в доставке семплов того или иного топика

ЛУЧШИЕ ПРАКТИКИ

Лучшие практики – 1/2

- Начните с описание модели данных, затем отразите ее на домены DDS, типы данных и топики
- Полностью и явно опишите типы данных: не используйте произвольные инкапсуляции
- Распределите подсистемы по доменам DDS
- Используйте типы данных с ключами

Лучшие практики – 2/2

- Опишите используемые QoS и их настройки. Лучше не использовать изменения настроек QoS в процессе функционирования, в больших системах (группах разработчиков) должны использоваться настройки QoS по умолчанию