OS Day 2023

Методология разработки прикладного или специализированного ПО на базе кибериммунного подхода

Иллюстрация применения для модели дрона-доставщика

Сергей Соболев
Старший архитектор по информационной безопасности,
KasperskyOS Community Development,
«Лаборатория Касперского»

KasperskyOS

План

Зачем?

Кибериммунность

Архитектура кибериммунного дрона и политики безопасности

Продукты с кибериммунитетом

Проблемы разработки

- Безопасность по умолчанию
- Нет требования – нет задачи
- Нефункциональное требование...
- Недостаток экспертизы
- Высокая стоимость специалистов

Проблемы кода

- Сложность решений
- Огромная кодовая база
- Высокая связность

- Чужой код
- Частые обновления
- Легаси

- Множество уязвимостей
- Вероятность НДВ
- Вероятность вредоносного кода

❌ Безопасность

Разработка

- Прозрачная методология Secure by Design, не требующая высоких ИБ-компетенций
- Процесс, ориентированный на компетенции членов команды

Код

- Понятные архитектурные требования
- Шаблоны проектирования
- Безопасность «из коробки»*

  *В случае использования KasperskyOS
- Обоснованное сокращение затрат на анализ и тестирование защищенности

✓ Безопасность

Кибериммунность

Методология разработки безопасных решений

KasperskyOS

Пожалуй, лучший инструмент для разработки кибериммунных решений

но не единственный

План

Зачем?

Кибериммунность

Архитектура кибериммунного дрона и политики безопасности

Кибериммунность

— это полноценная методология, содержащая весь набор атрибутов (дисциплины, практики и методы)

Как построить решение,
которому можно доверять,
из компонентов, большинству
из которых доверять нельзя?

Кибериммунность

— Как?

1. Требования к **архитектуре**
2. Требования к **процессу**

Кибериммунность

❶ Архитектура

Secure by Design:

Система должна быть спроектирована так, чтобы быстро обосновать ее безопасность

Три фундаментальных принципа:

- Изоляция
- Контроль
- Минимизация TCB*

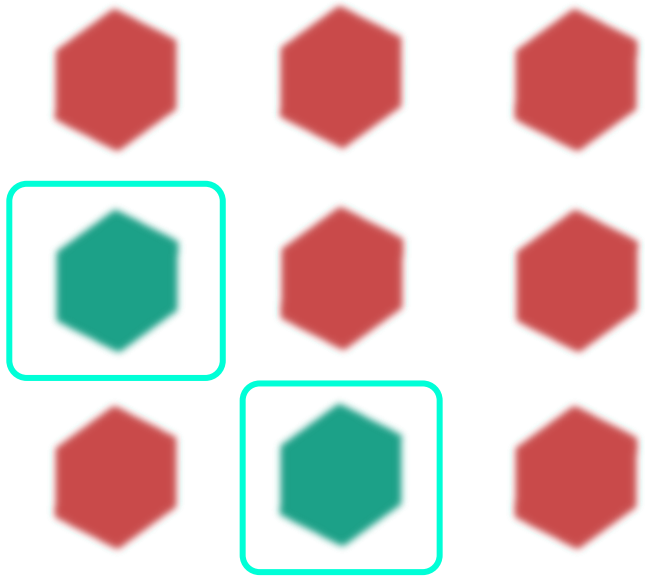
* Trusted Code Base

Кибериммунность

❶ Архитектура

Три принципа → ZeroTrust

Вместо анализа и противостояния огромной поверхности атаки заботимся о небольшой поверхности защиты



Кибериммунность

❷ Процесс

1. Концепция *безопасности* продукта
2. Цели и предположения безопасности
3. Архитектура
4. Разработка и тестирование
5. Моделирование угроз
6. Верификация

«Гибкость в процессе, жесткость в артефактах»

План

Зачем?

Кибериммунность

Архитектура кибериммунного дрона и политики безопасности

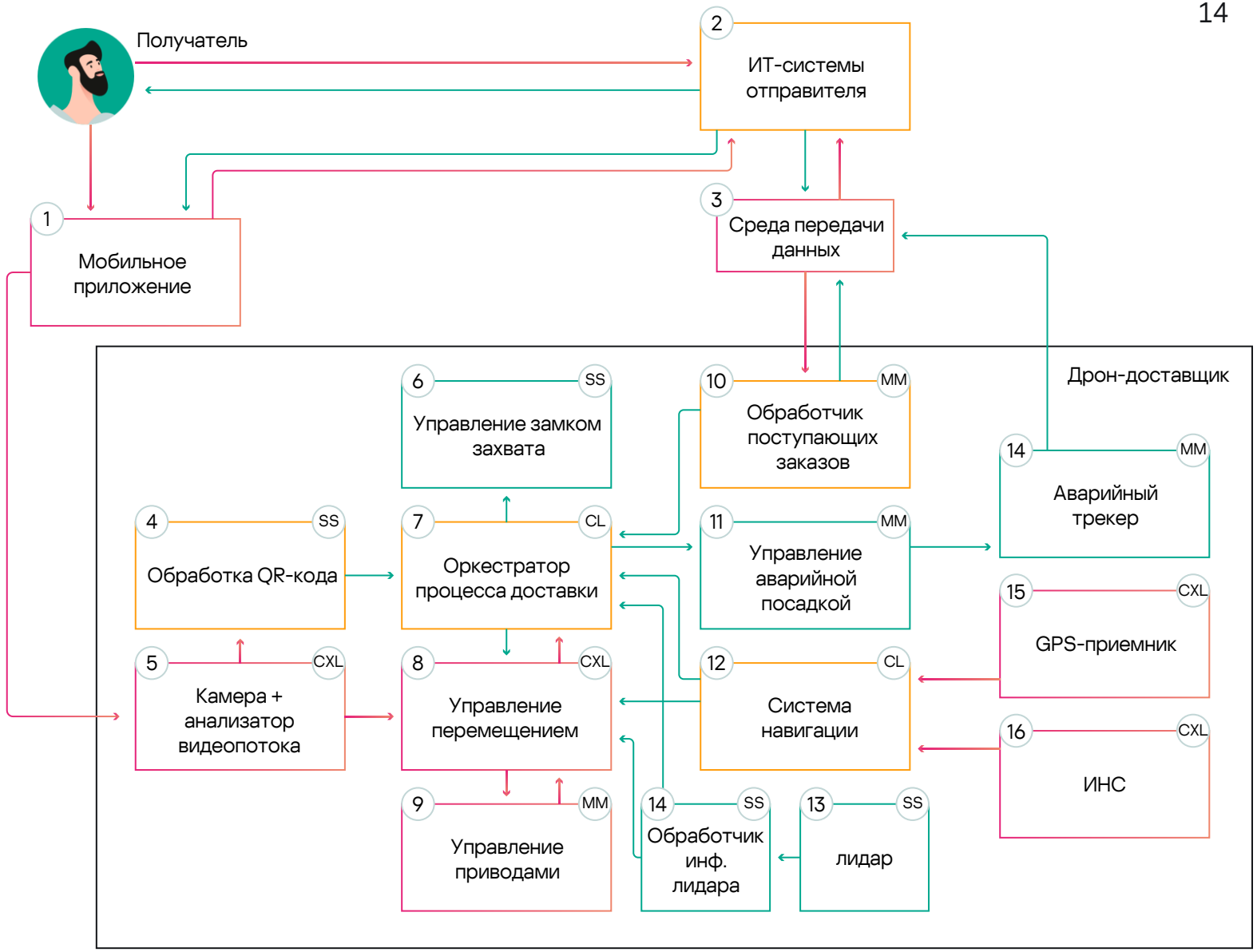
Кибериммунный дрон-доставщик

Цели безопасности

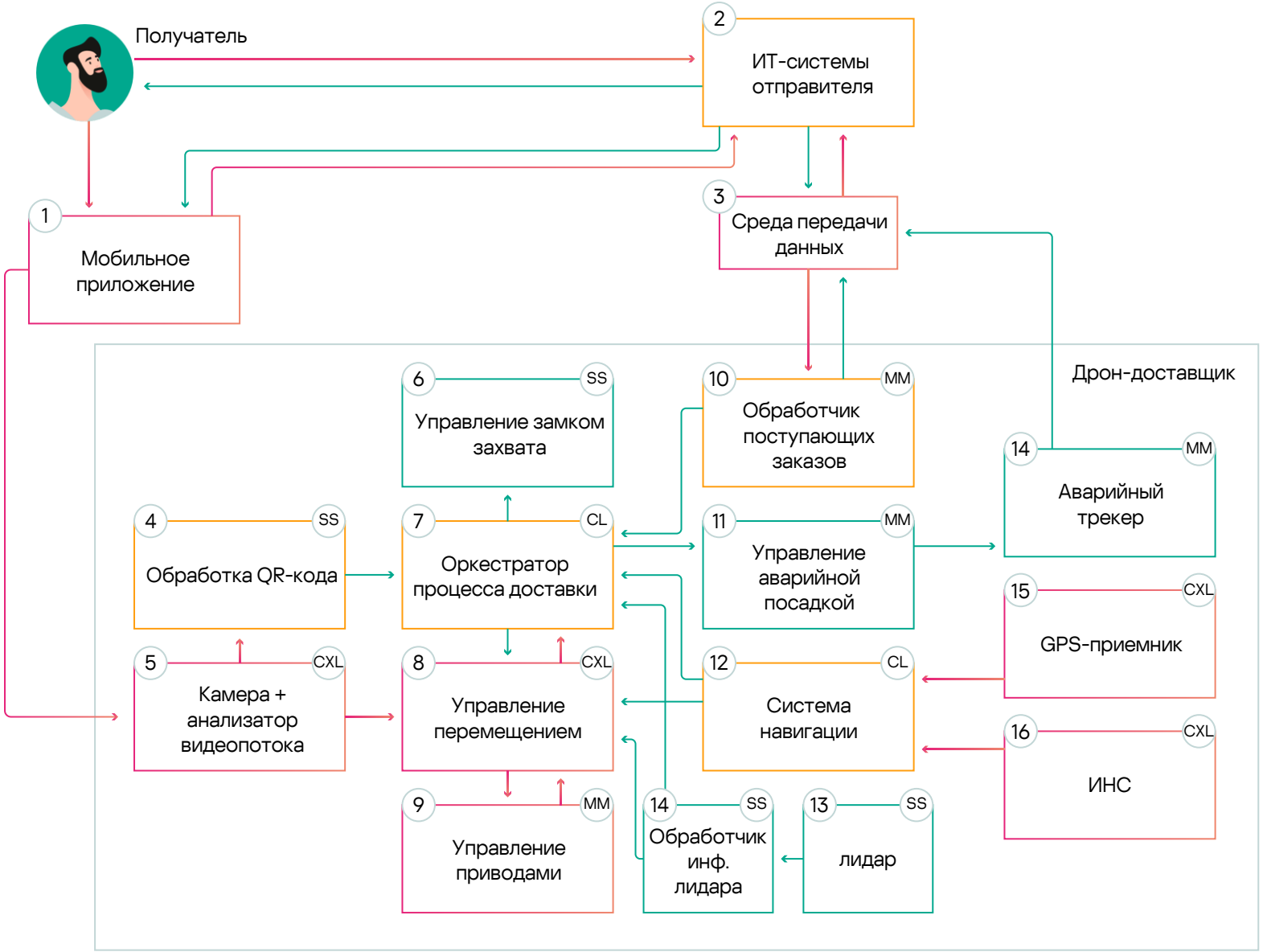
1. Только аутентичные и авторизованные получатели получают заказ
2. Для полета используются только разрешенные для полетов районы и эшелоны
3. ИТ-система отправителя в любой ситуации имеет достоверную информацию о положении дрона
4. На маршруте дрон всегда находится на безопасном расстоянии от стационарных объектов

Предположения безопасности

1. ИТ-системы отправителя защищены
2. Мобильное приложение получателя нельзя считать доверенным



Кибериммунный дрон-доставщик

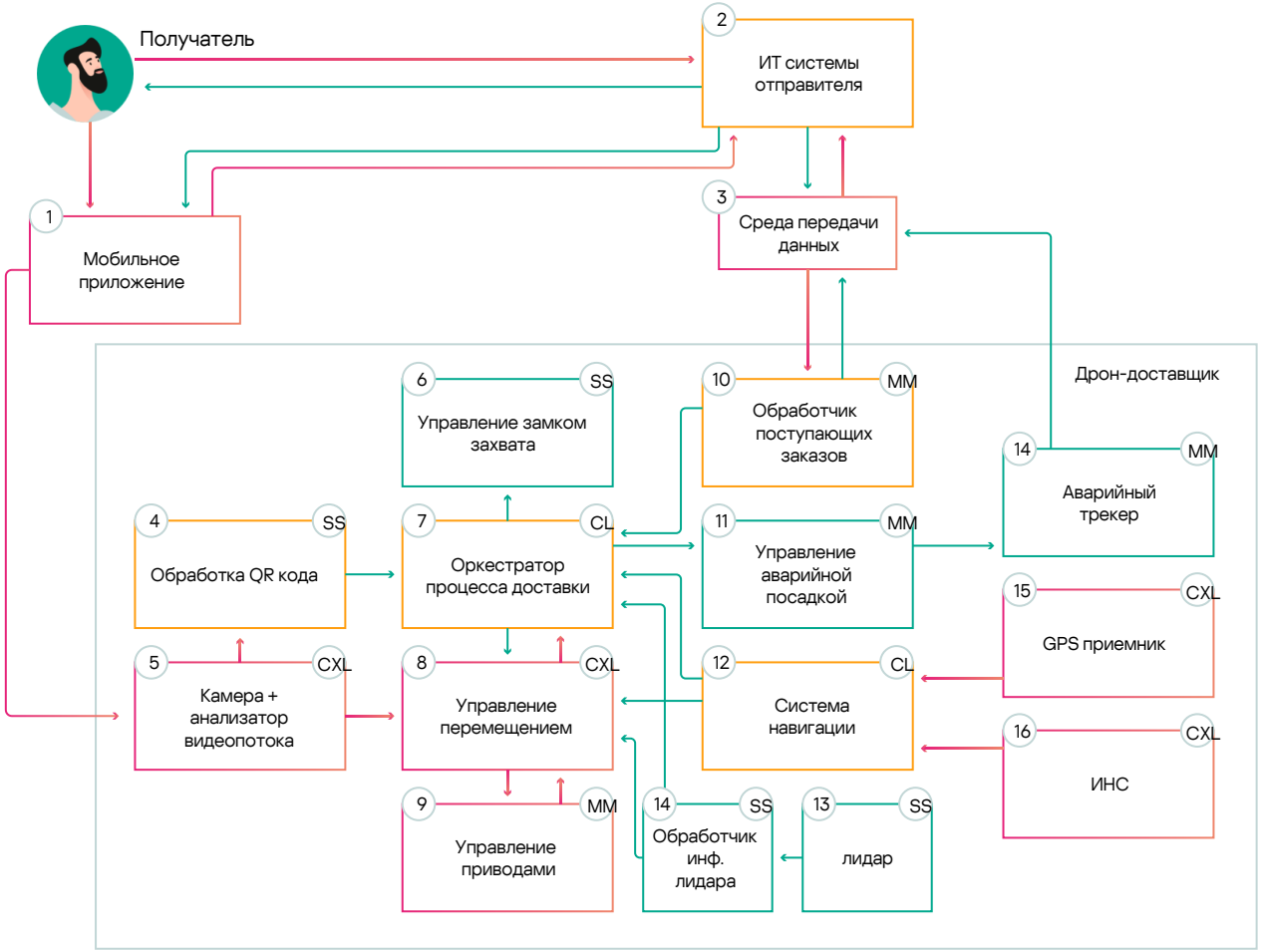


Легенда

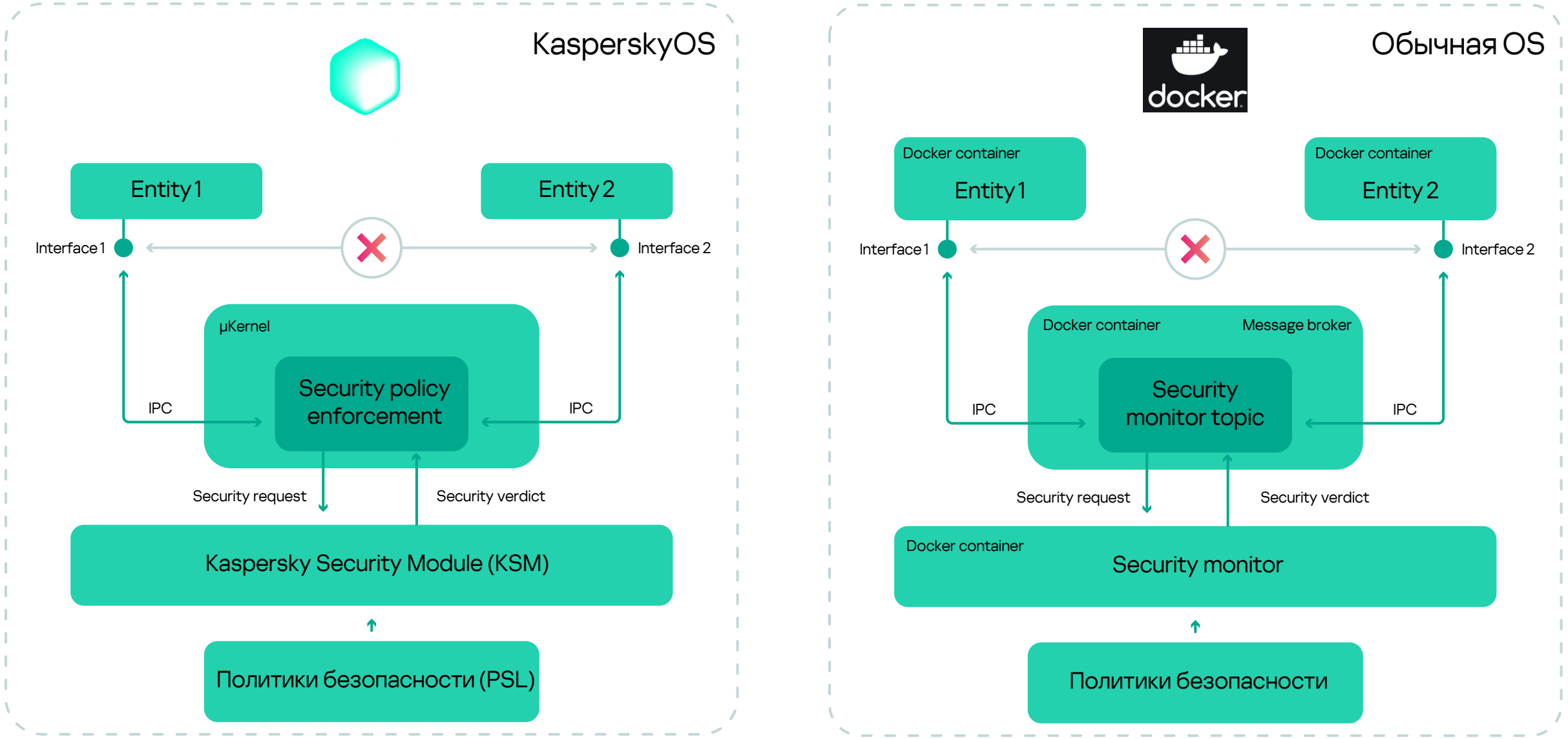
- Недоверенная сущность
- Доверенная сущность
- Доверенная сущность, повышающая целостность данных
- Высокоцелостные данные
- Низкоцелостные данные

Иллюстрация обоснования уровня доверия сущностей

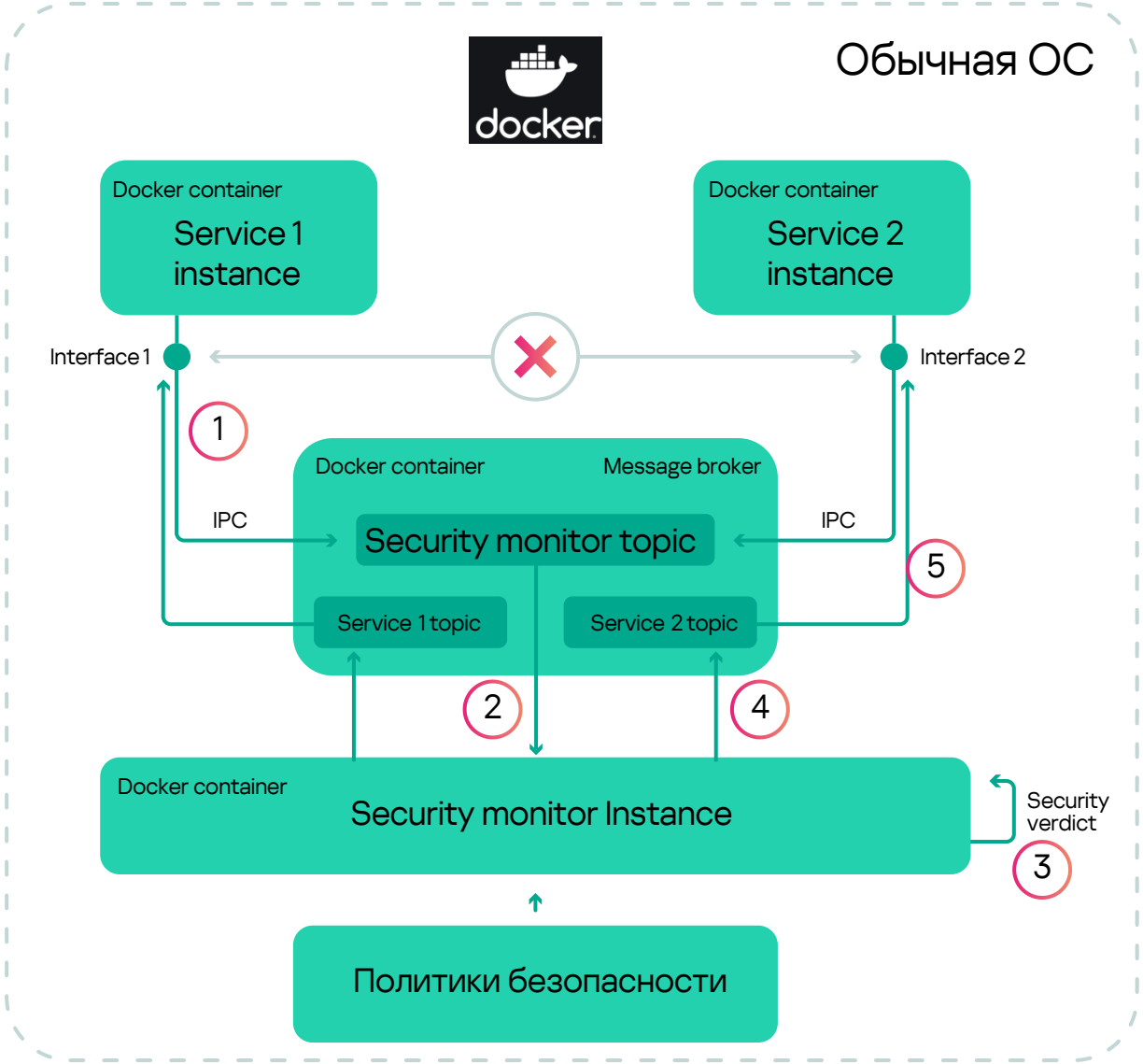
Цель Безопасности	Сущность	Уровень доверия	Обоснование
Для полета используются только разрешенные для полетов районы и эшелоны	Система навигации	Доверенная, повышающая целостность данных	Задача системы навигации собирать данные от GPS и ИНС, и повышать целостность этих данных для дальнейшего использования в системе управления перемещением и оркестрацией процесса доставки. Также система навигации будет отвечать за компенсацию ошибки ИНС. Т.к. система навигации отвечает за позиционирование в нашей системе, она должна быть доверенной, иначе мы не сможем достичь ЦБ 2 и 3. Т.к. она работает с низкоцелостными GPS данными, она должна быть «желтой» - повышающей целостность данных - сущностью.



Продуктовый кибериммунитет на KasperskyOS, а учиться проектировать можно и на контейнерах



Последовательность передачи сообщений Service1 -> Service2



Итоги

- Методология кибериммунности предлагает конкретные методы и практики для каждого этапа жизненного цикла разработки ПО
- Конструктивная безопасность реализуется для обеспечения целей безопасности
- Только такой код переводится в разряд доверенного, без которого достижение целей безопасности невозможно
- Есть способы качественной оценки объема доверенного кода на уровне архитектуры системы
- Предложенная архитектура дрона-доставщика обеспечивает достижение выбранных целей безопасности без необходимости делать бОльшую часть кода доверенной
- Кибериммунный подход к разработке универсален и не зависит от конкретных технологий и инструментов. Он снижает риски эксплуатации уязвимостей.
- Кибериммунный подход следует изучать и внедрять для разработки прикладного и специализированного ПО

Спасибо!

Обучение кибериммунному подходу к разработке
https://t.me/learning_cyberimmunity

@LEARNING_CYBERIMMUNITY

Дополнительная информация по теме (теория, задачи, решения)
https://kas.pr/ba1m

Сергей Соболев

Старший архитектор по информационной безопасности, KasperskyOS Community Development

Sergey.P.Sobolev@kaspersky.com

KasperskyOS