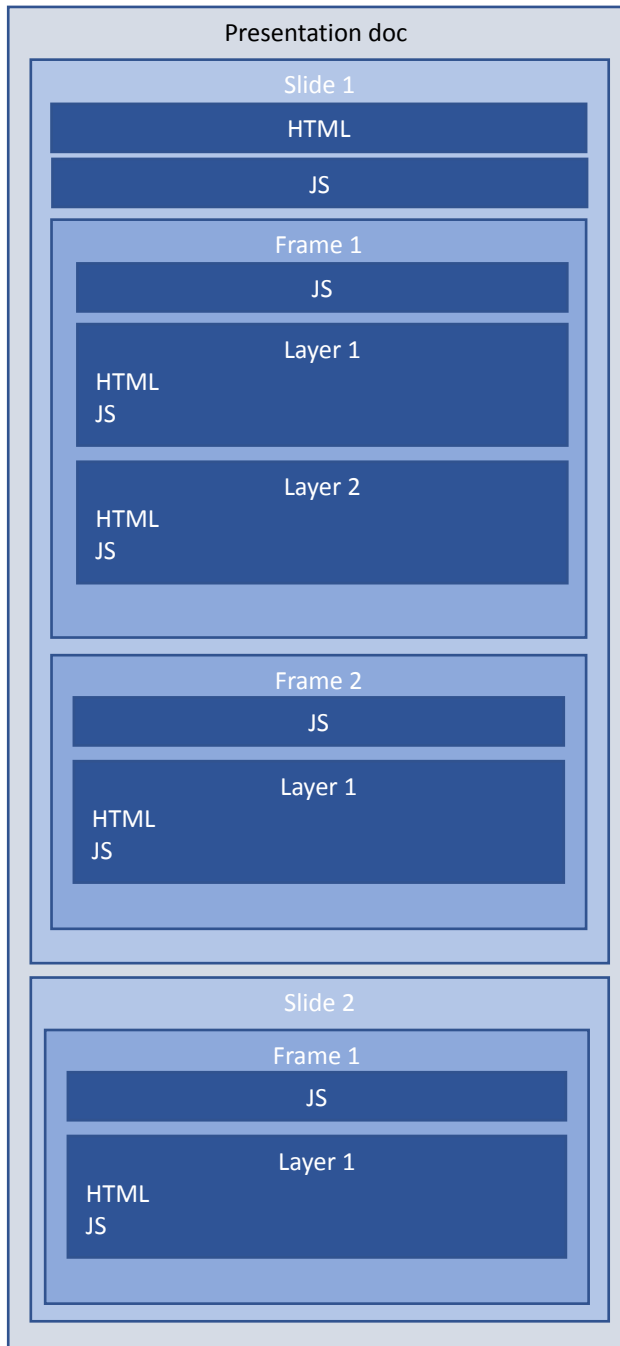
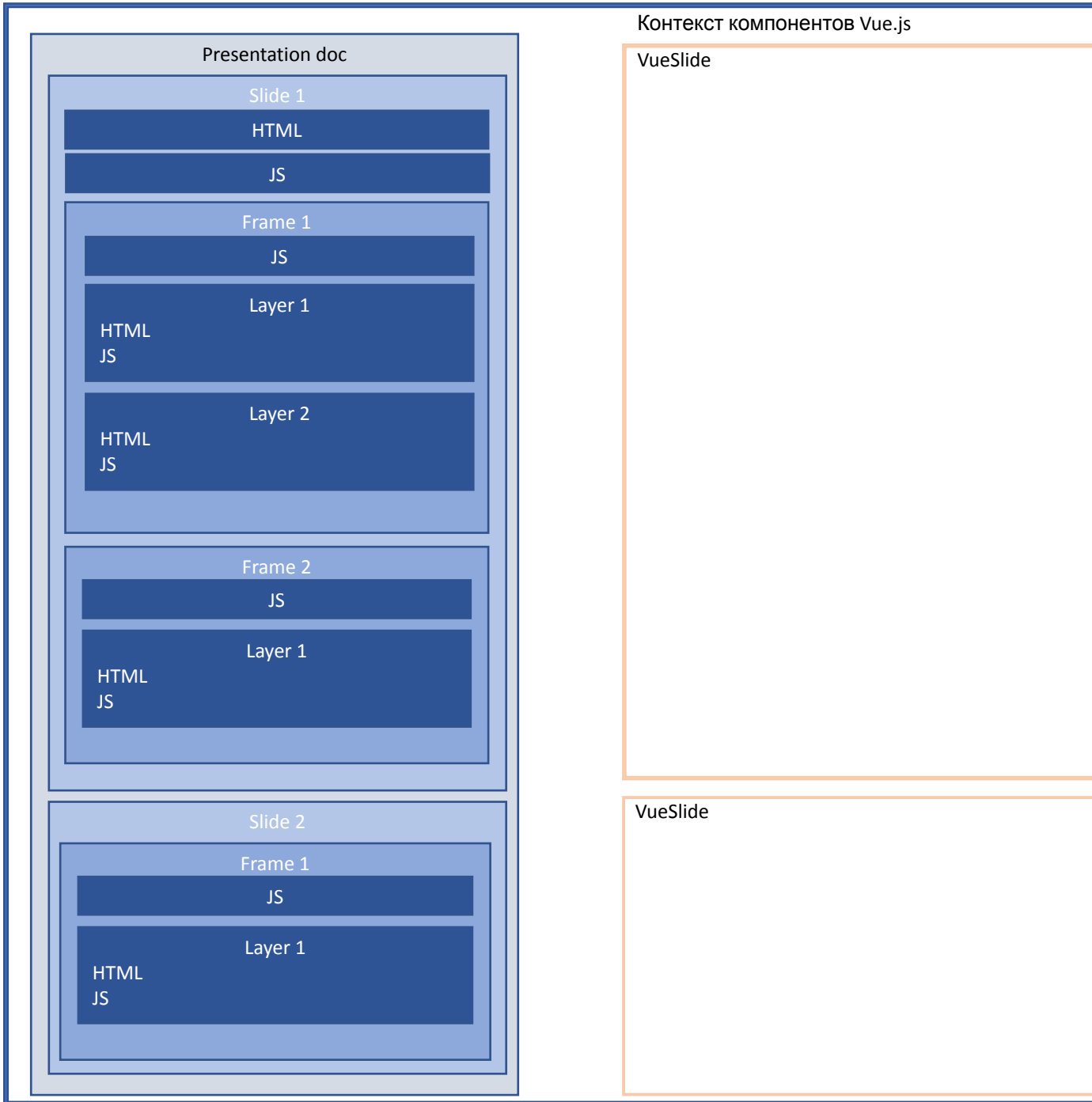


Архитектура приложения «Медиашоу»

ООО «Севен Скейл», Таганрог, 2022



В презентации Слайды
В слайдах Фреймы
Во фреймах Слои

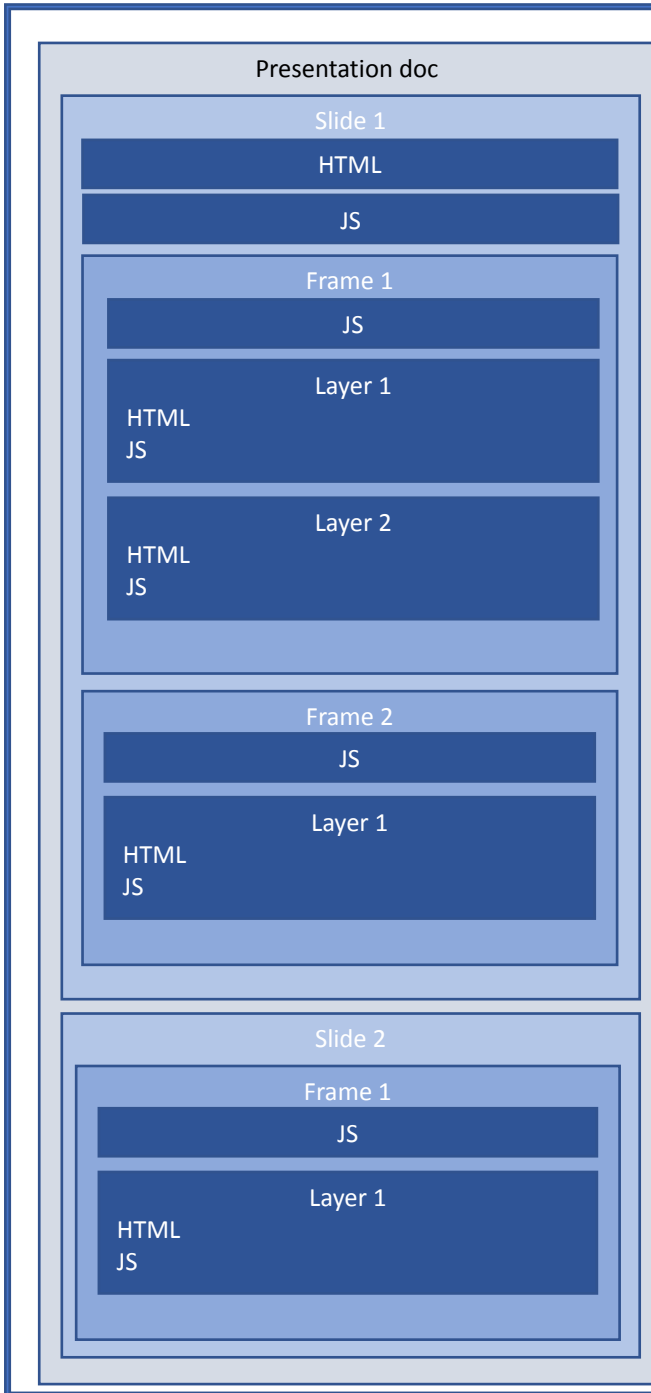


Контекст КОМПОНЕНТОВ Vue.js

VueSlide

Синяя рамка – контекст Vue.js
Персиковая рамка – Компоненты VueSlide, являющиеся рамочными с точки зрения контента и кода VueSlide служат в роли контейнеров и связующих элементов между двумя контекстами Vue.js и Vanilla JS

VueSlide



Контекст компонентов Vue.js

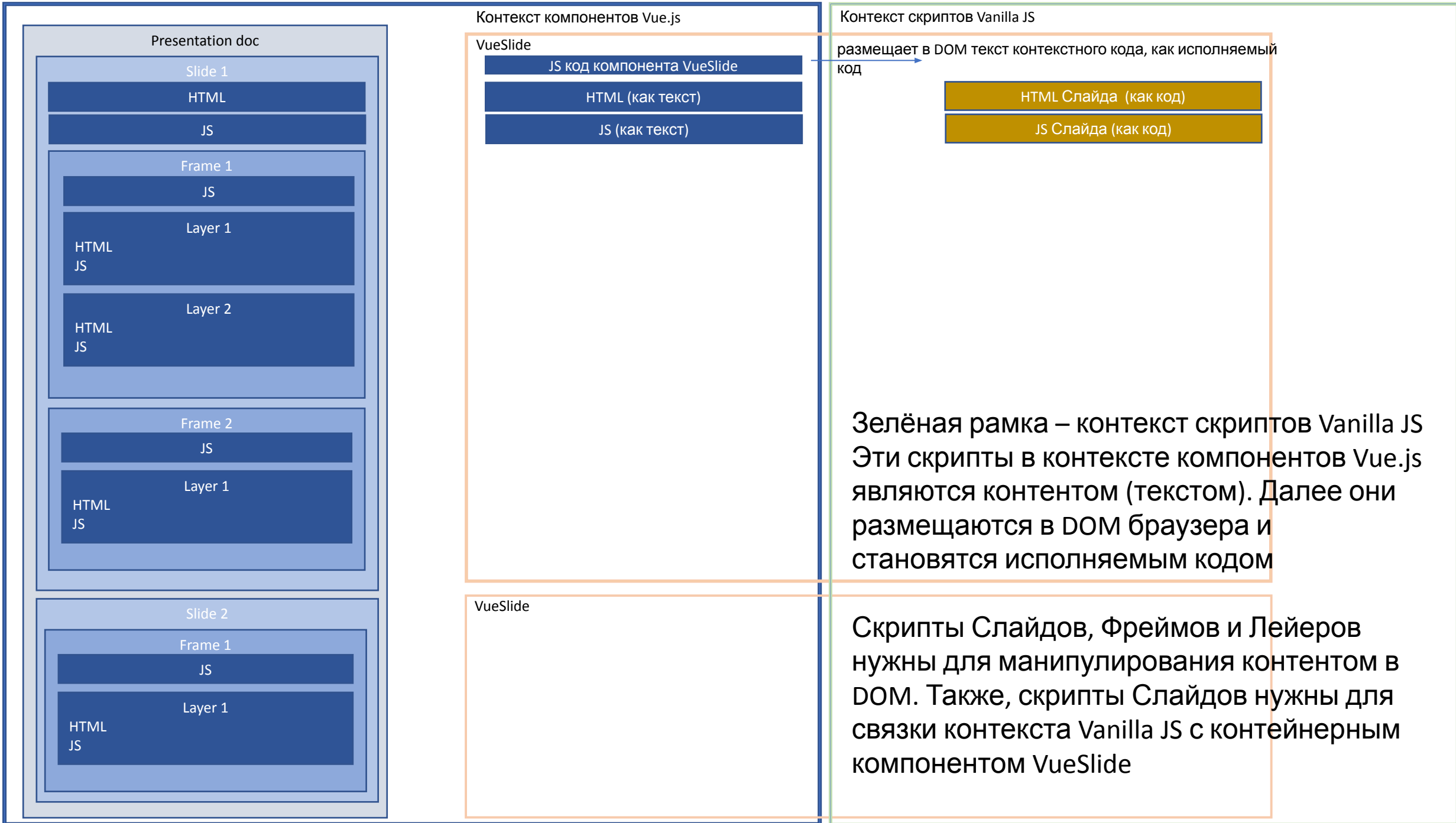
VueSlide

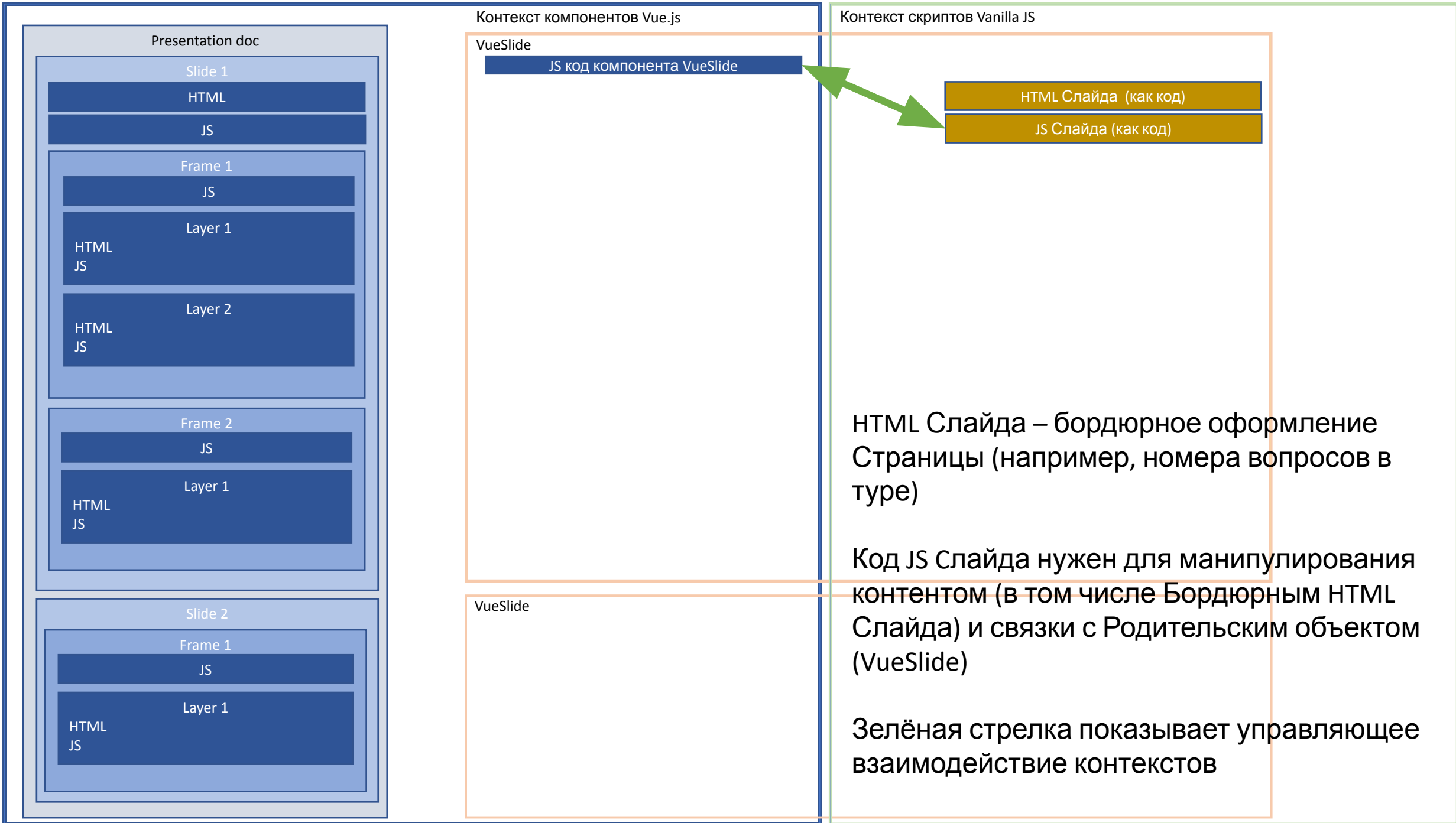
VueSlide

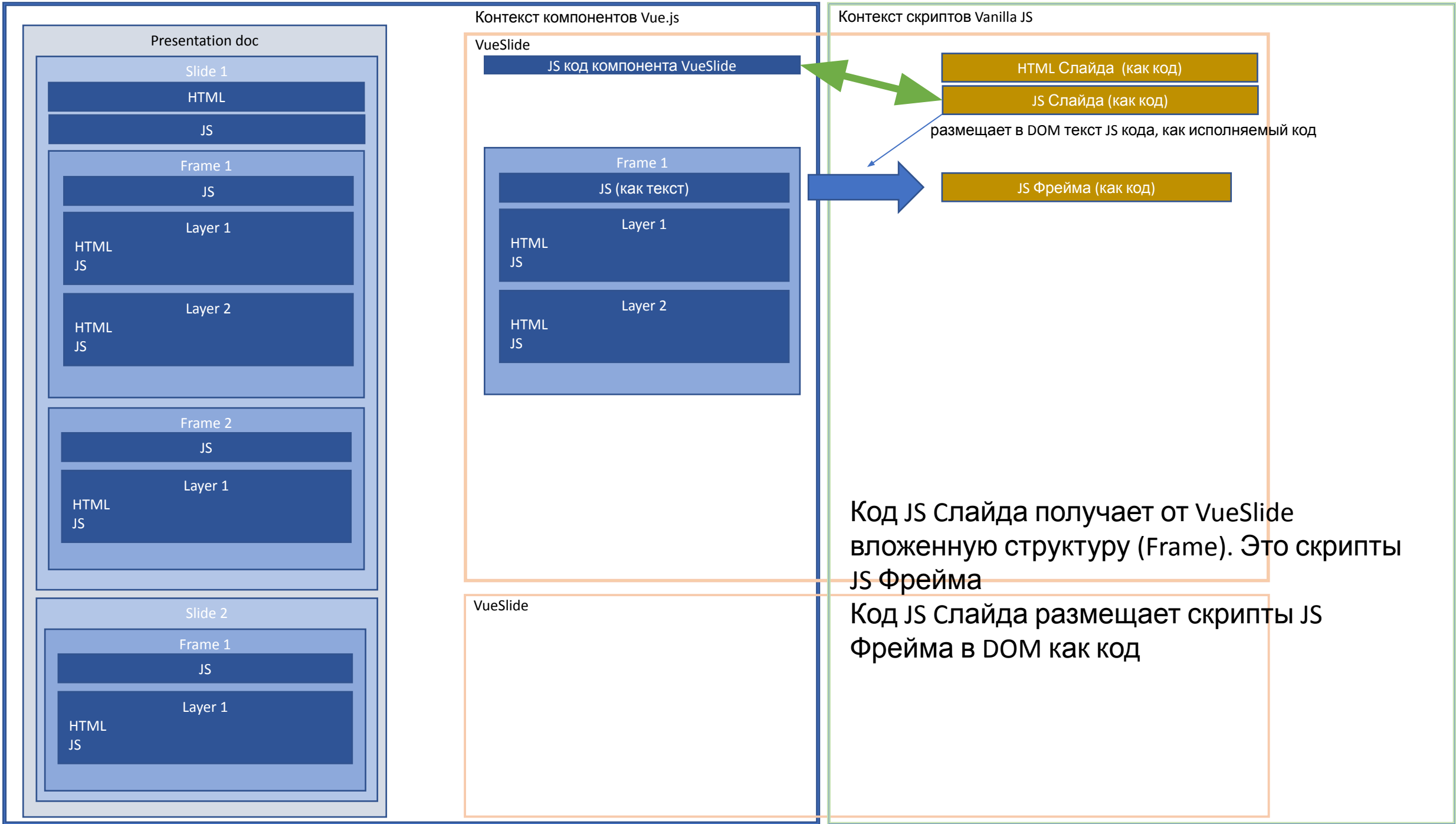
Контекст скриптов Vanilla JS

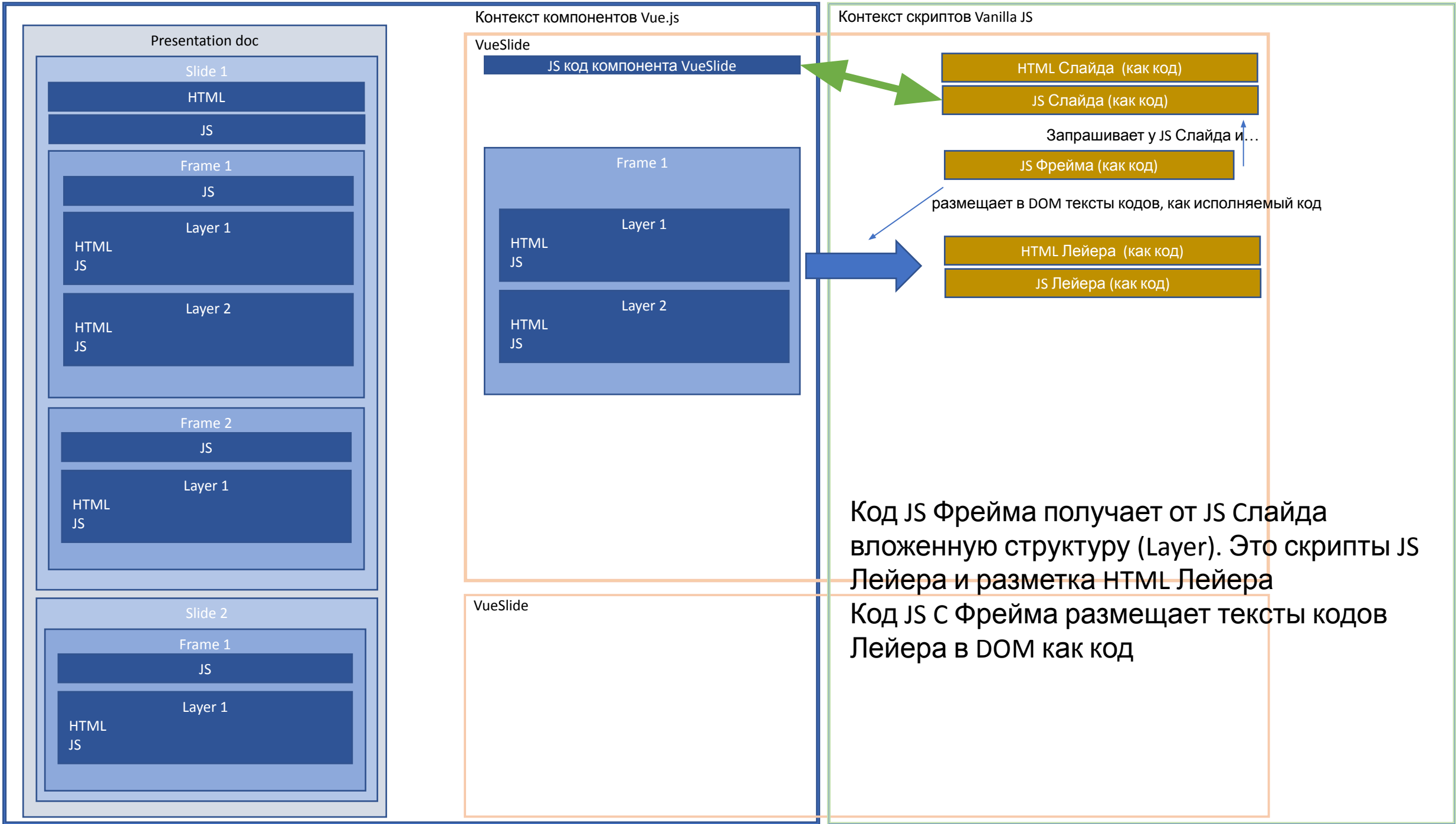
Зелёная рамка – контекст скриптов Vanilla JS
Эти скрипты в контексте компонентов Vue.js являются контентом (текстом). Далее они размещаются в DOM браузера и становятся исполняемым кодом

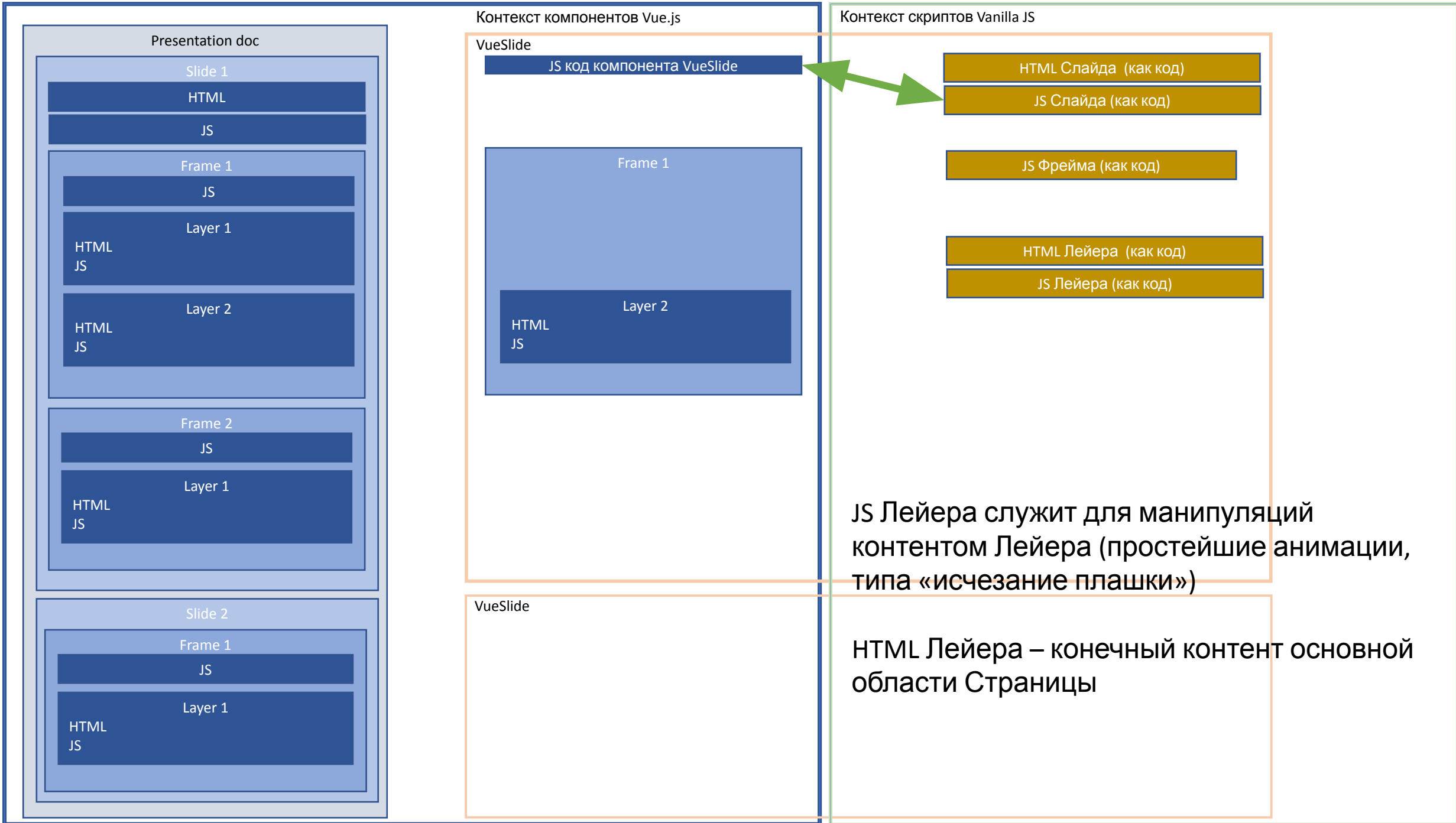
Скрипты Слайдов, Фреймов и Лейеров нужны для манипулирования контентом в DOM. Также, скрипты Слайдов нужны для связки контекста Vanilla JS с контейнерным компонентом VueSlide

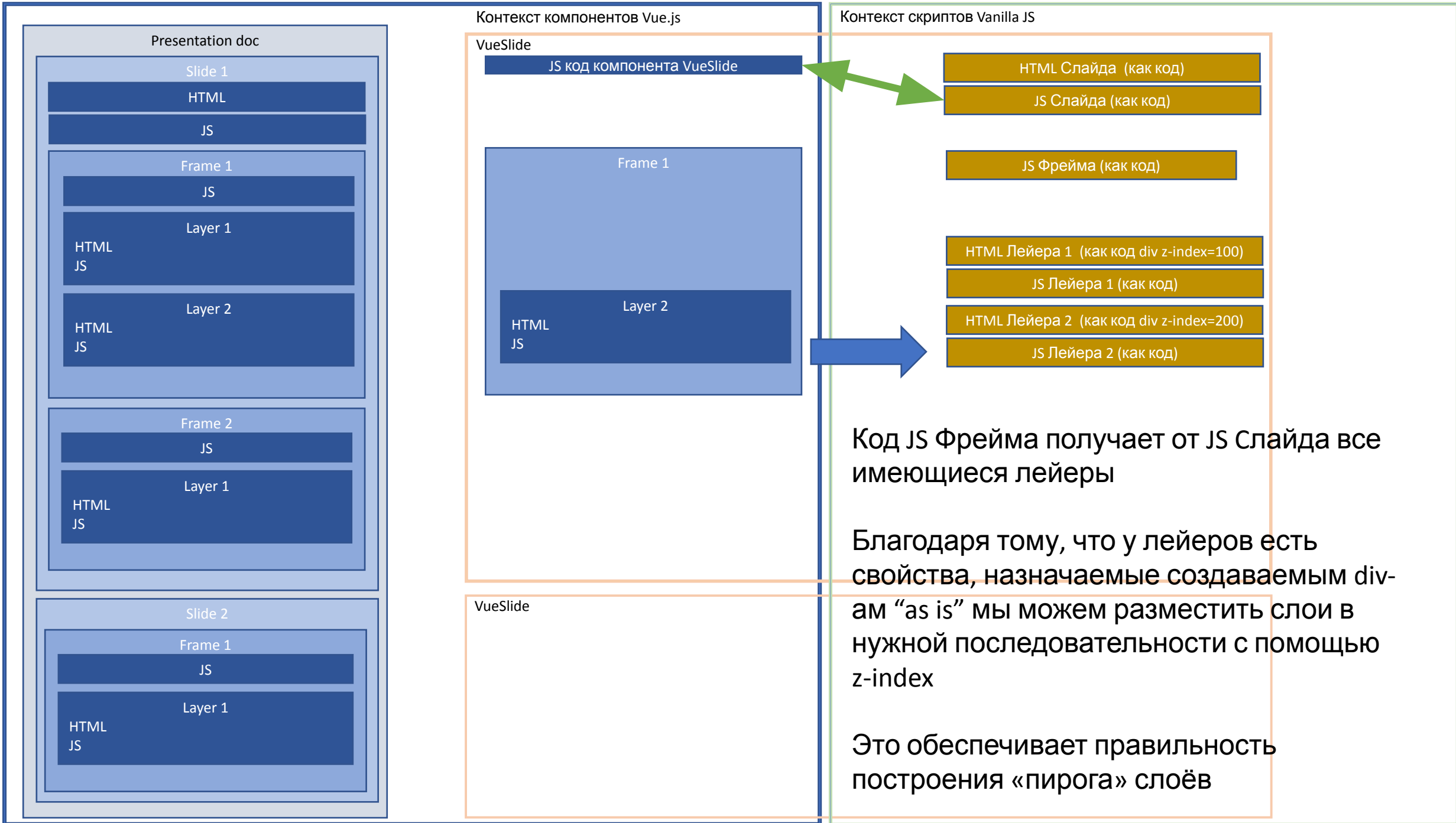












Контекст компонентов Vue.js

Контекст скриптов Vanilla JS

Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

VueSlide

JS код компонента VueSlide

Frame 1

Layer 2

HTML
JS

VueSlide

HTML Слайда (как код)

JS Слайда (как код)

JS Фрейма (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

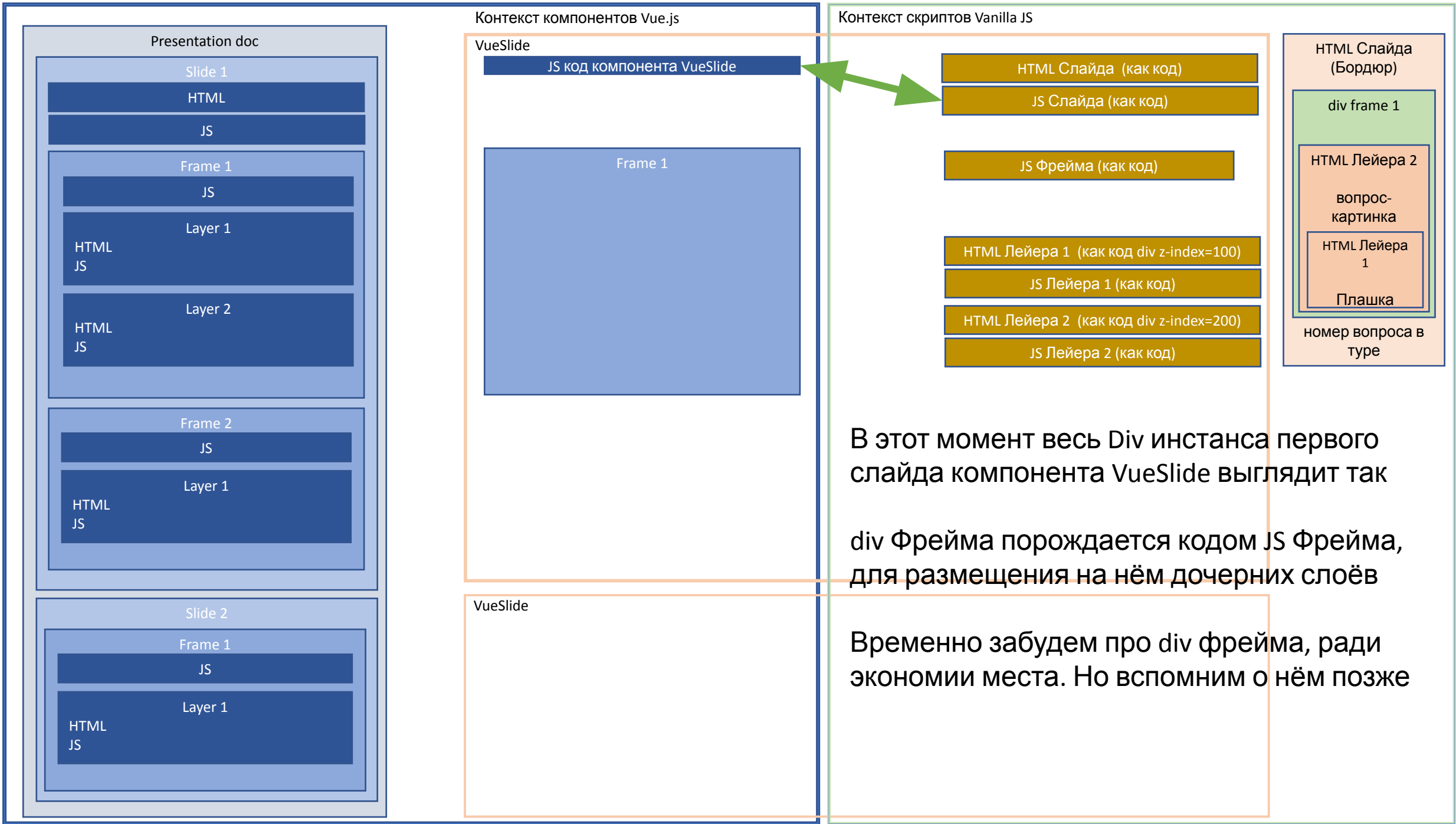
HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

Код JS Фрейма получает от JS Слайда все имеющиеся лейеры

Благодаря тому, что у лейеров есть свойства, назначаемые создаваемым div-ам "as is" мы можем разместить слои в нужной последовательности с помощью z-index

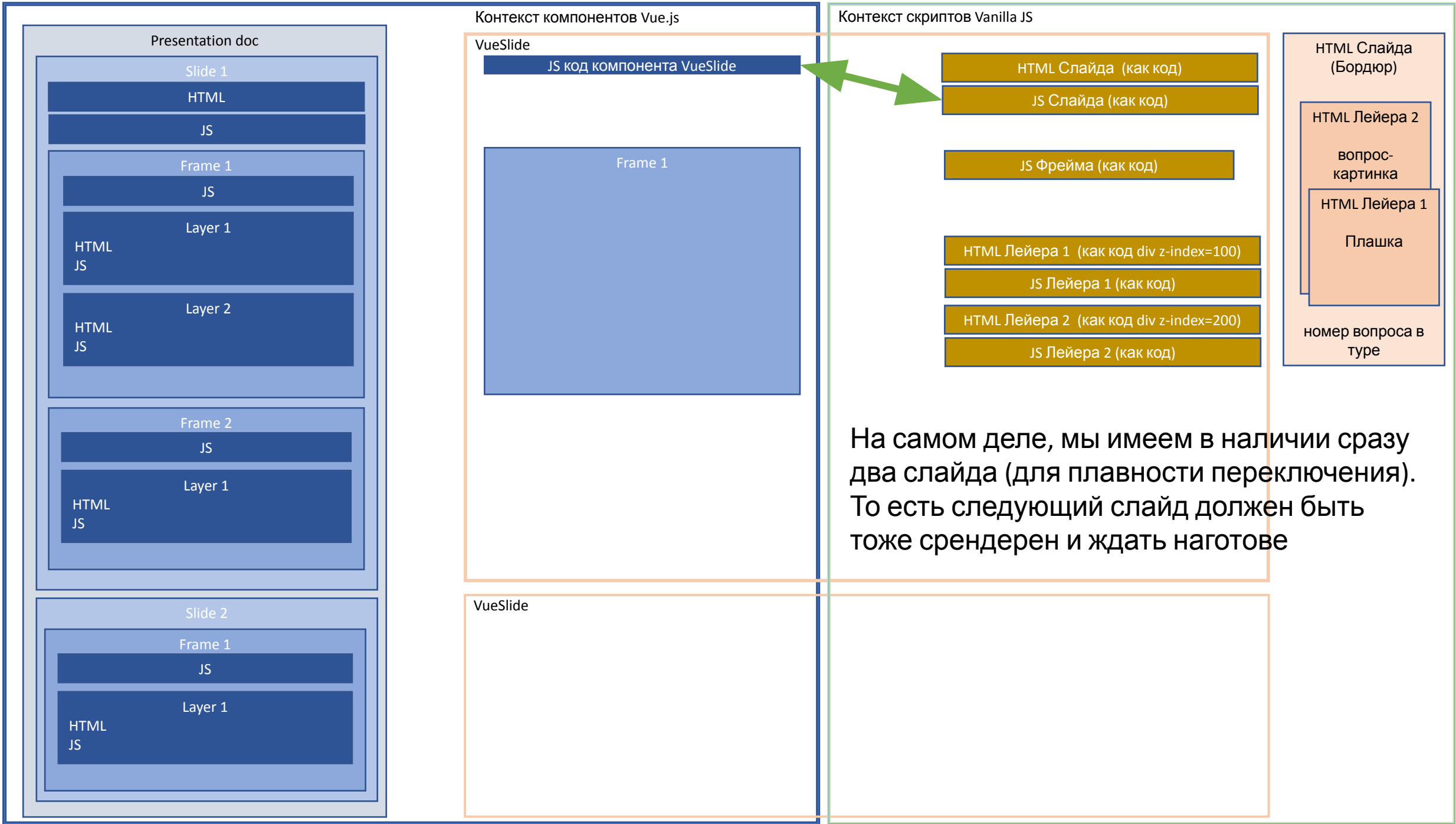
Это обеспечивает правильность построения «пирога» слоёв

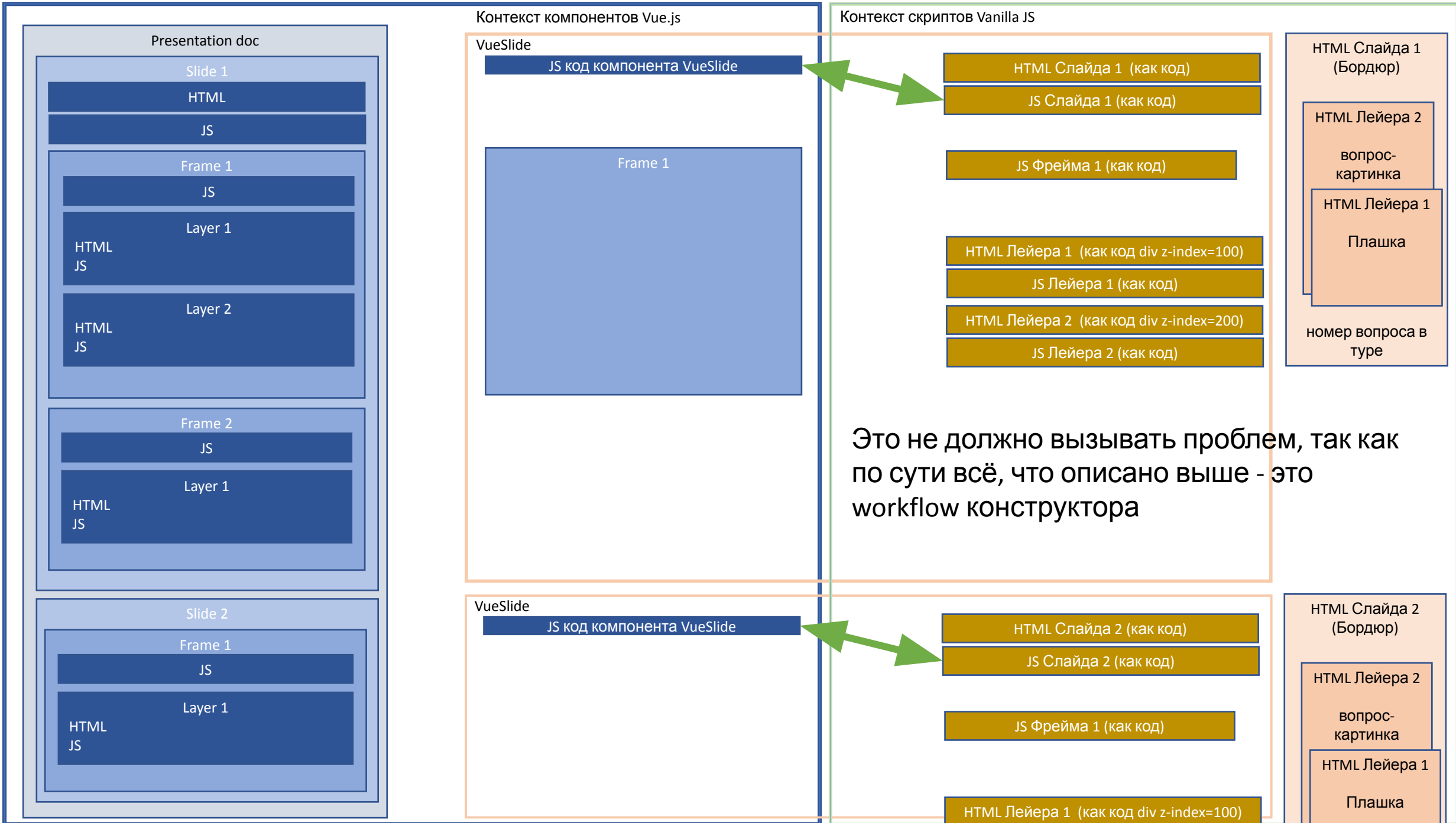


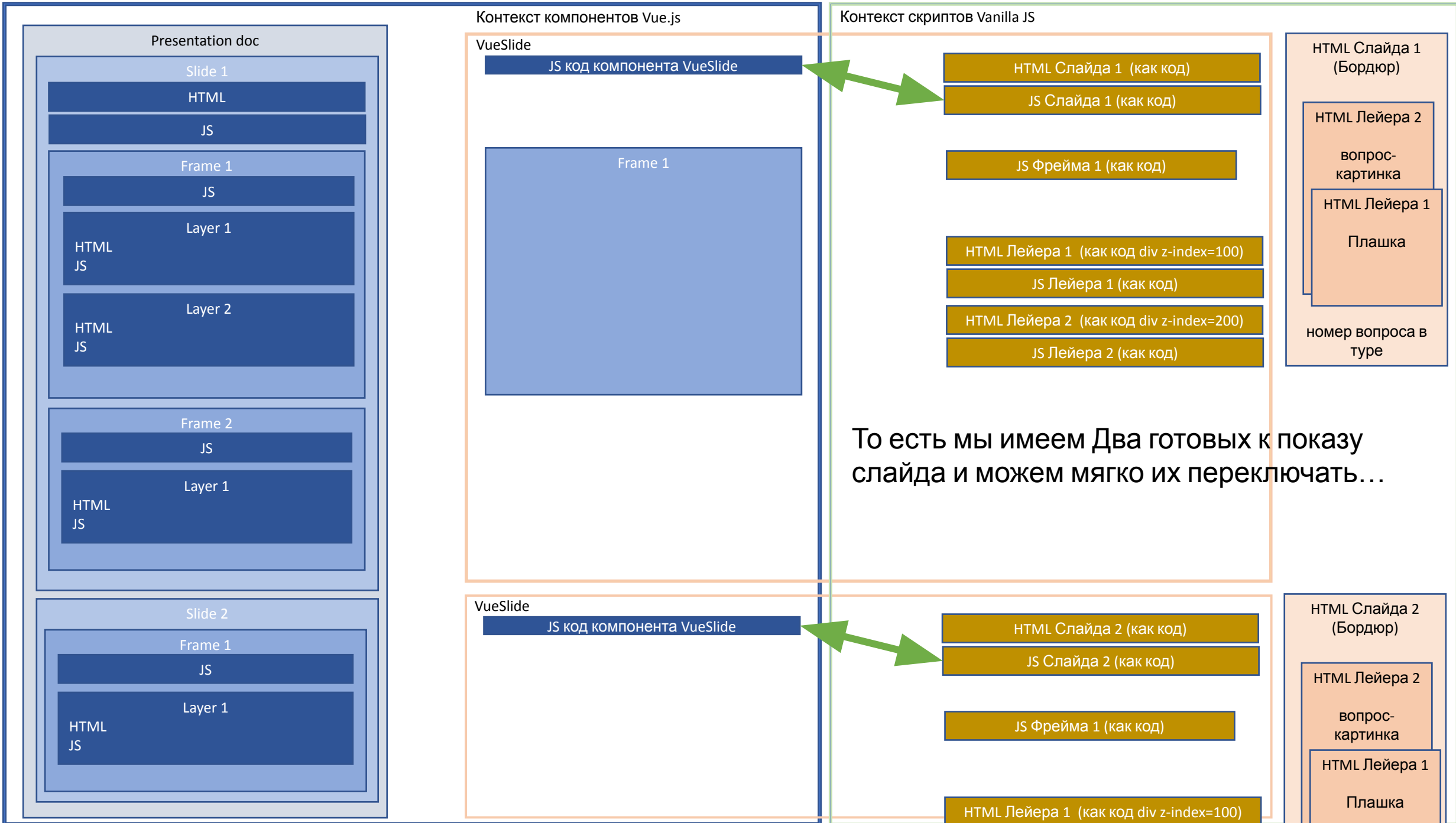
В этот момент весь Div инстанса первого слайда компонента VueSlide выглядит так

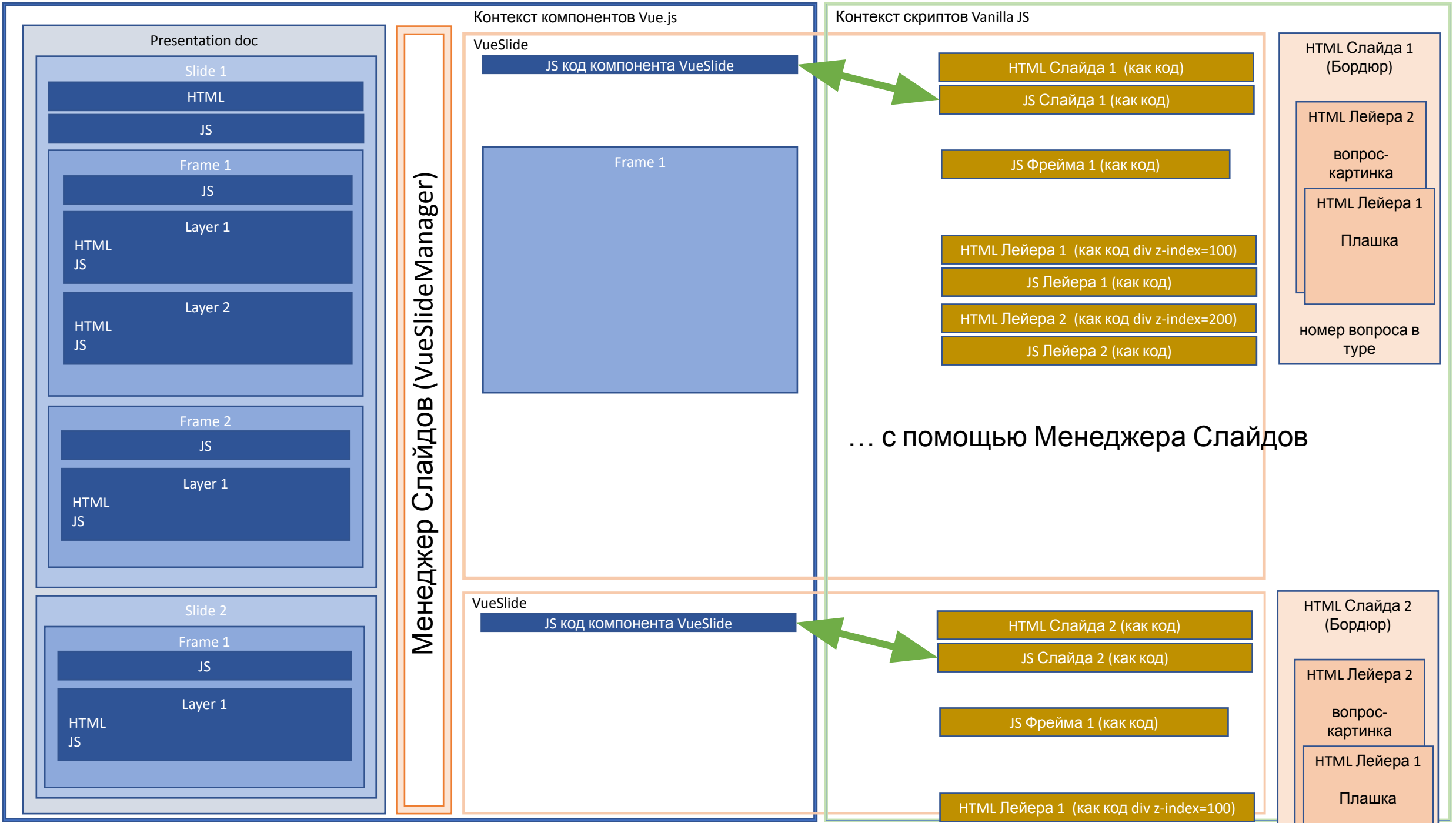
div Фрейма порождается кодом JS Фрейма, для размещения на нём дочерних слоёв

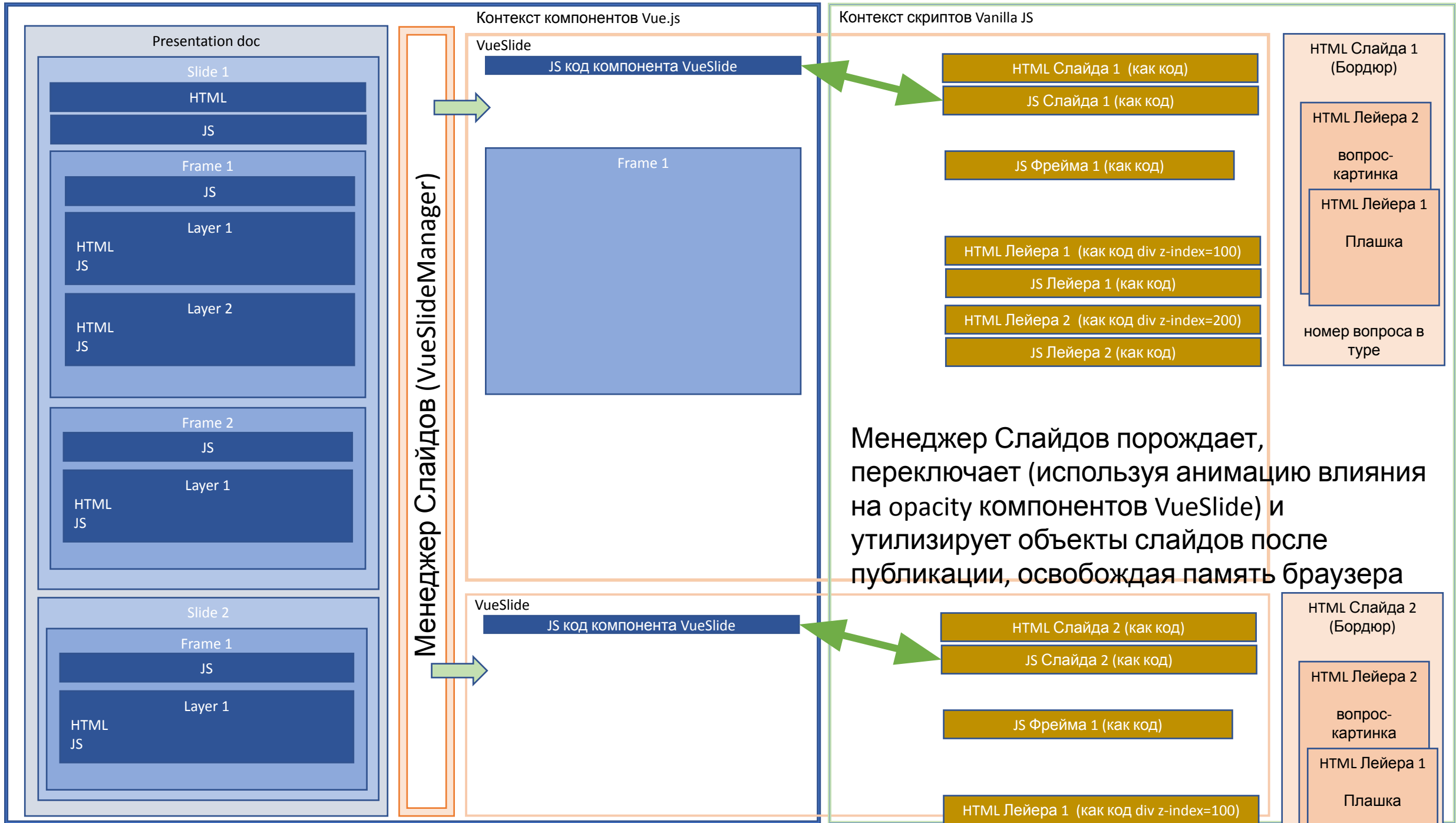
Временно забудем про div фрейма, ради экономии места. Но вспомним о нём позже

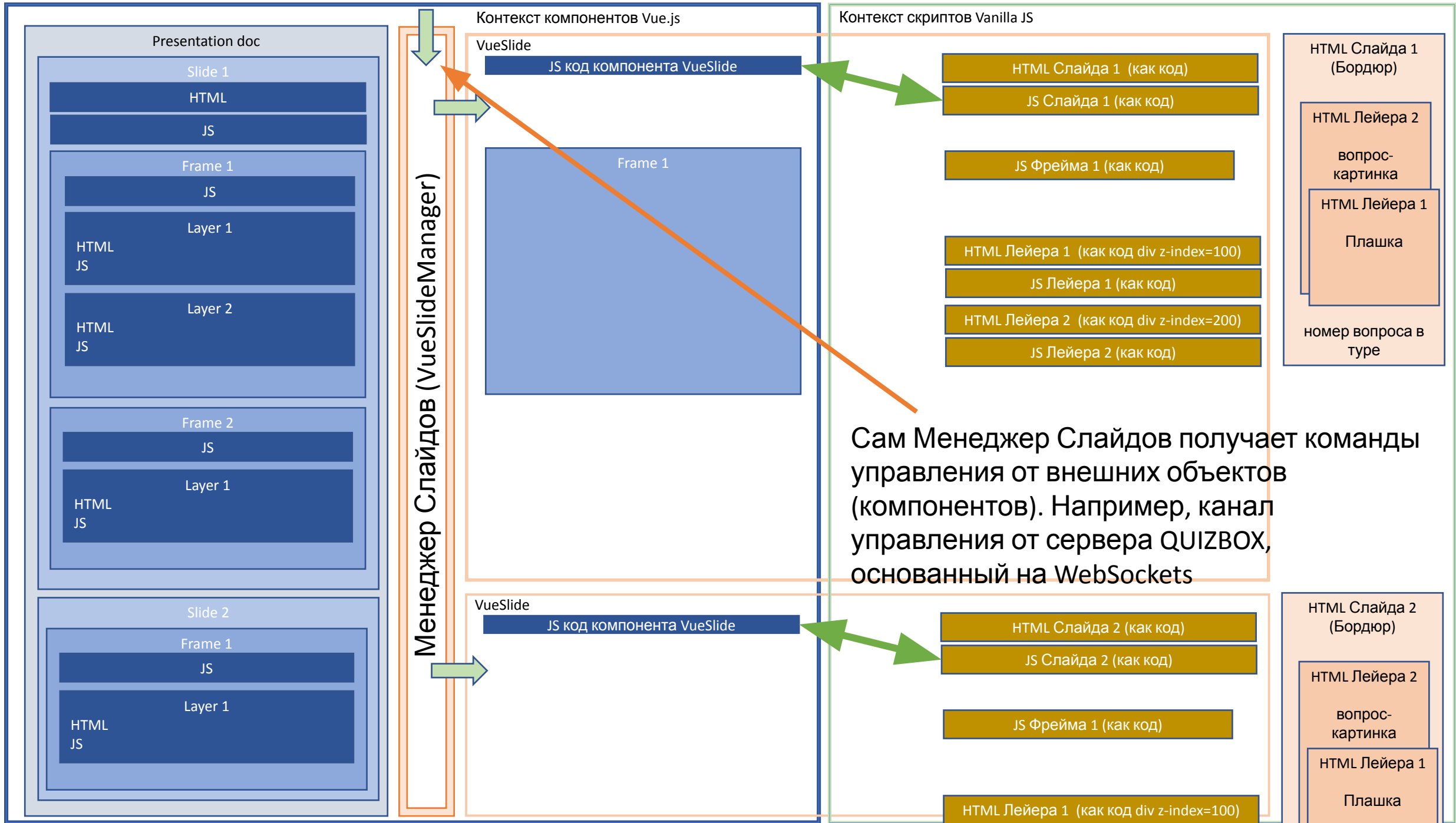


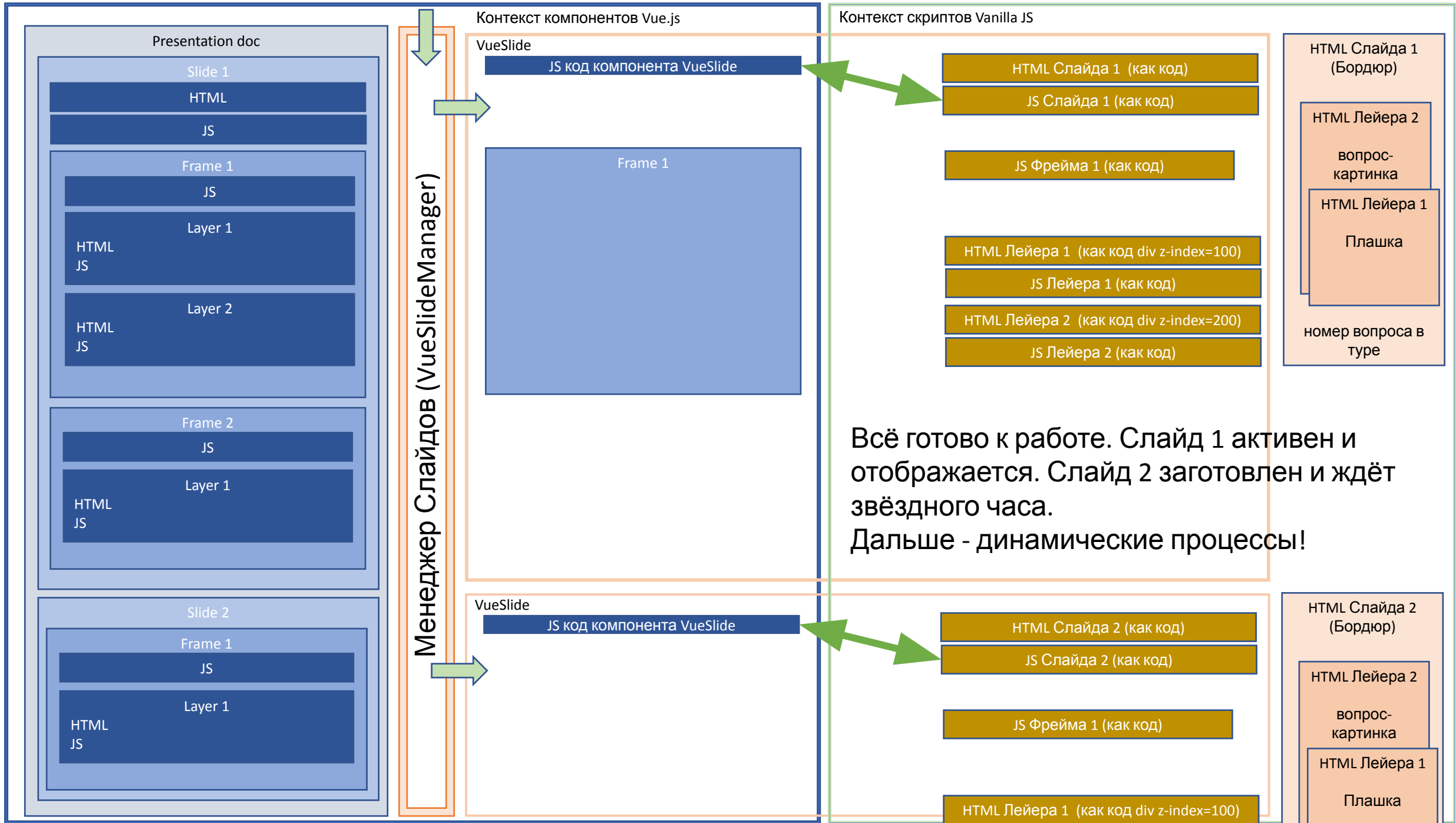












Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 1

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

HTML Слайда 1 (Бордюр)

HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка

номер вопроса в туре

Всё готово к работе. Слайд 1 активен и отображается. Слайд 2 заготовлен и ждёт звёздного часа. Дальше - динамические процессы!

HTML Слайда 2 (Бордюр)

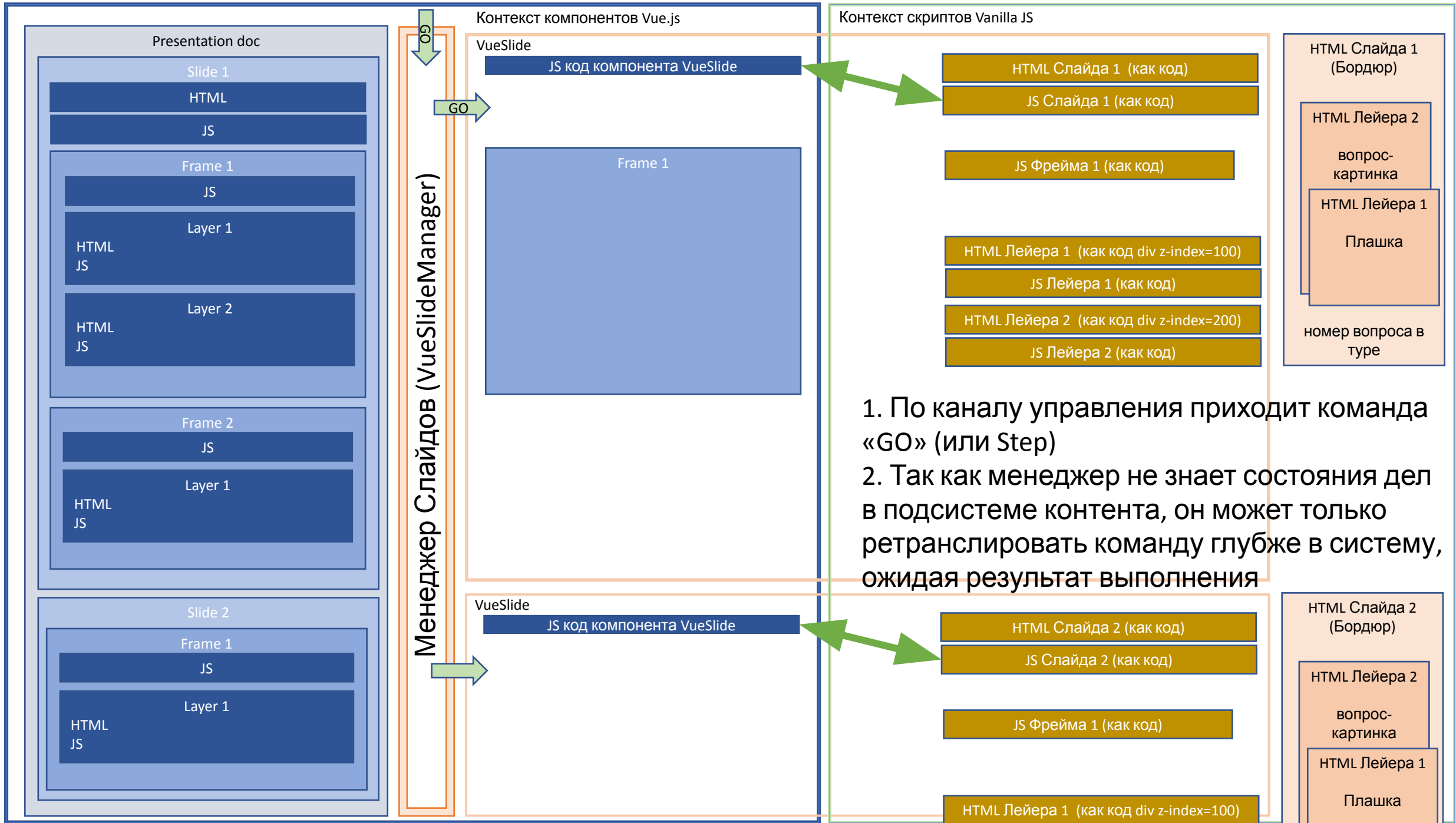
HTML Лейера 2

вопрос-картинка

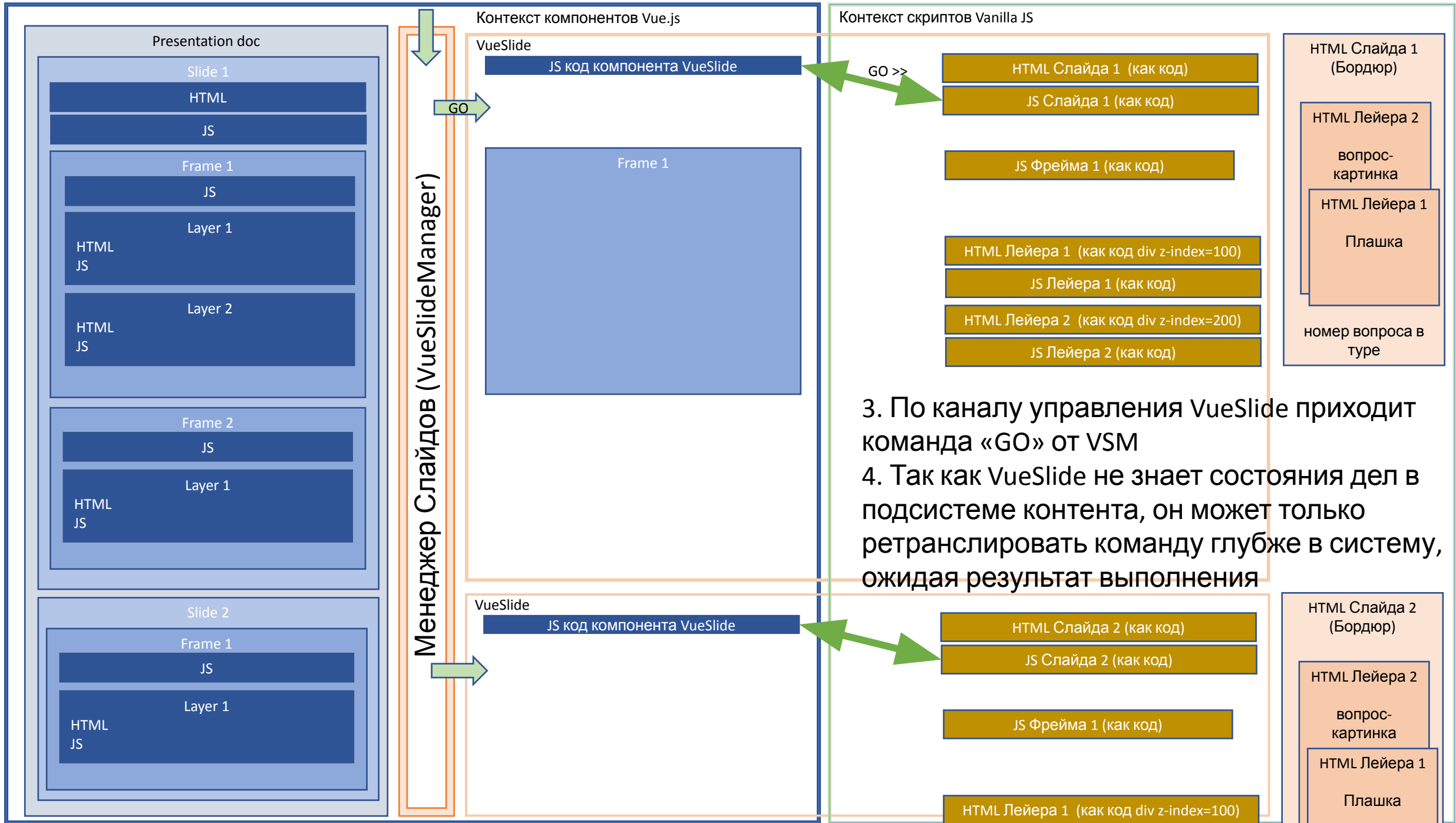
HTML Лейера 1

Плашка

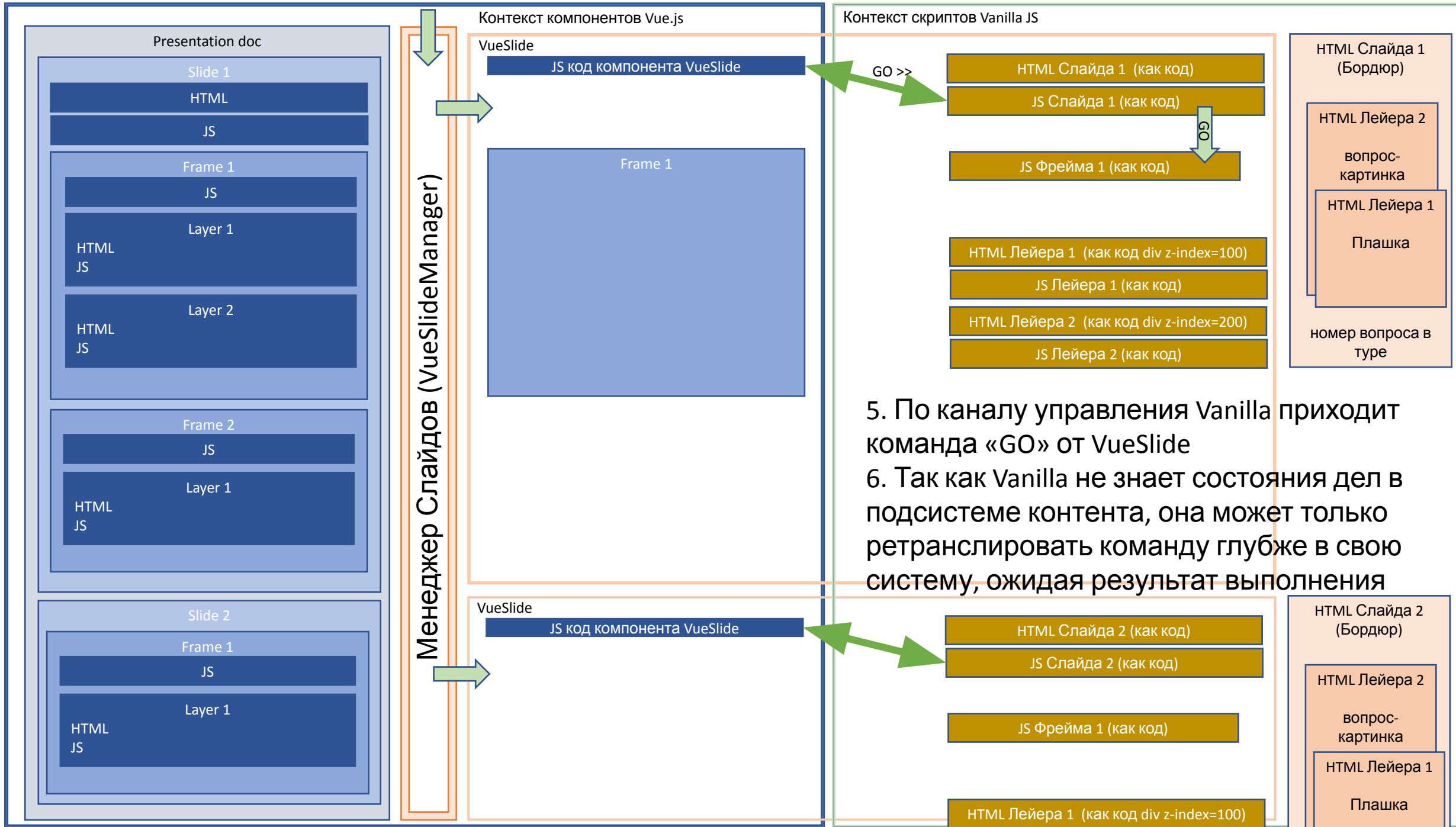
Менеджер Слайдов (VueSlideManager)



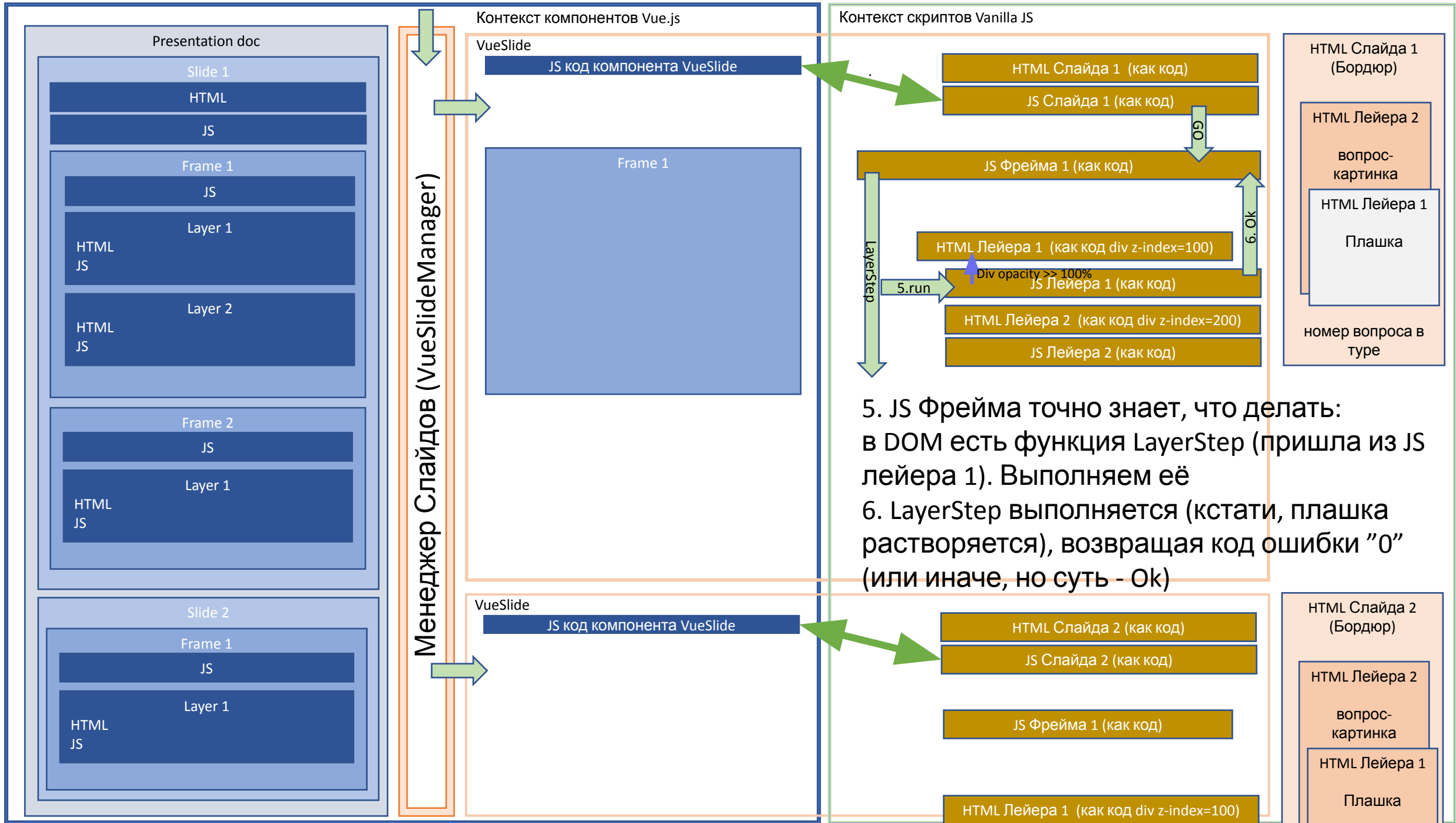
1. По каналу управления приходит команда «GO» (или Step)
2. Так как менеджер не знает состояния дел в подсистеме контента, он может только ретранслировать команду глубже в систему, ожидая результат выполнения



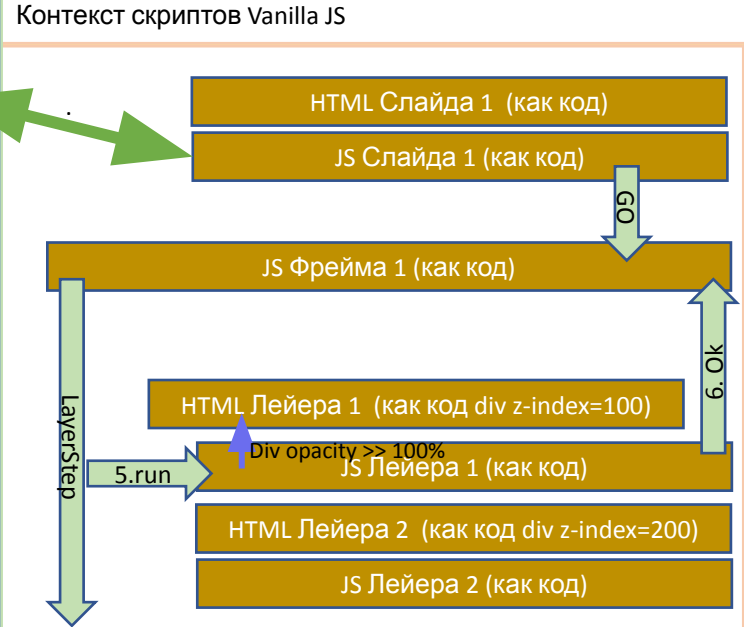
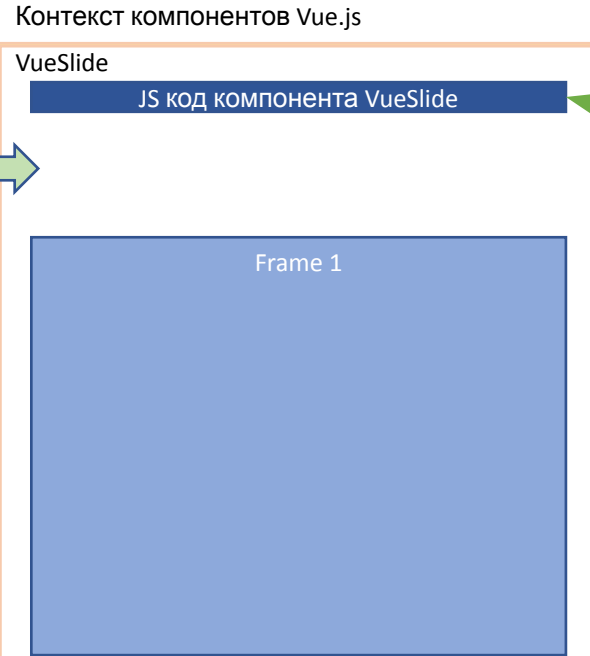
3. По каналу управления VueSlide приходит команда «GO» от VSM
 4. Так как VueSlide не знает состояния дел в подсистеме контента, он может только ретранслировать команду глубже в систему, ожидая результат выполнения



5. По каналу управления Vanilla приходит команда «GO» от VueSlide
 6. Так как Vanilla не знает состояния дел в подсистеме контента, она может только ретранслировать команду глубже в свою систему, ожидая результат выполнения

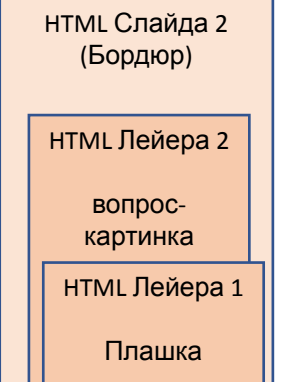
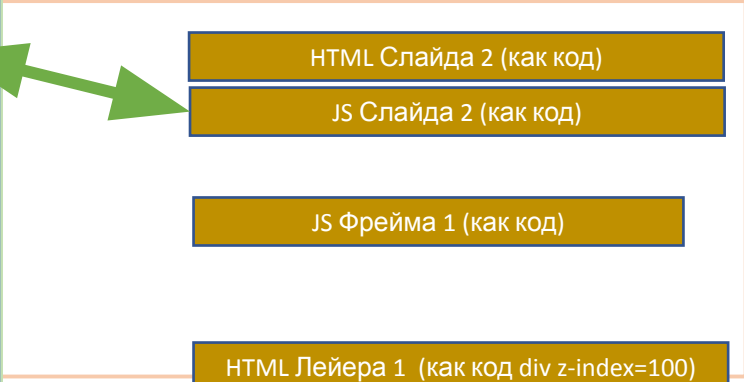
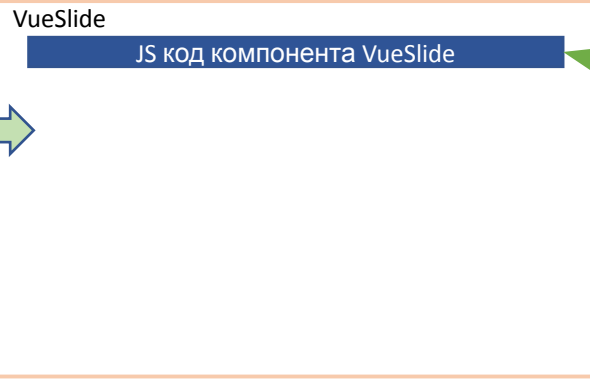


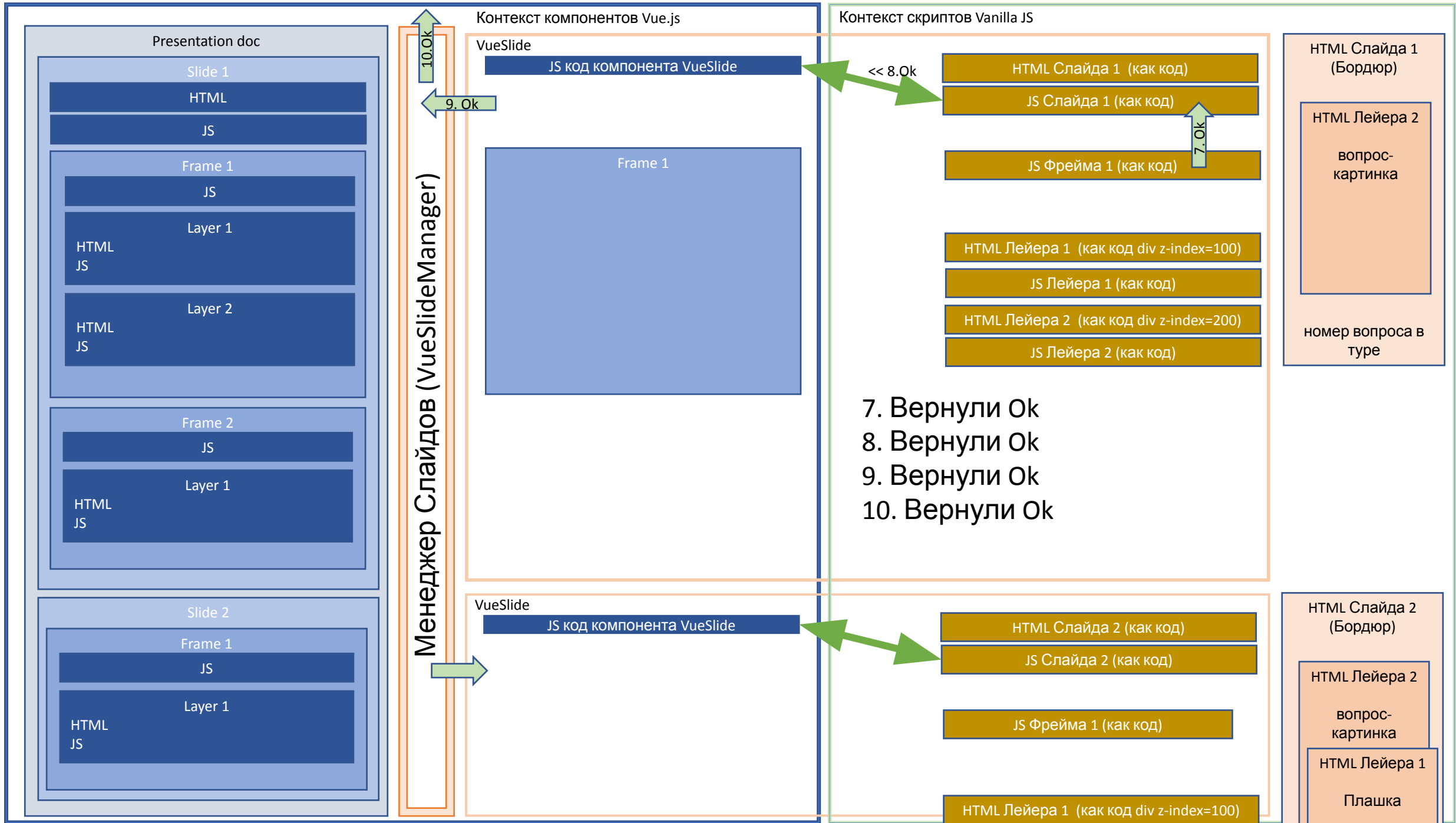
Менеджер Слайдов (VueSlideManager)



5. JS Фрейма точно знает, что делать: в DOM есть функция LayerStep (пришла из JS лейера 1). Выполняем её

6. LayerStep выполняется (кстати, плашка растворяется), возвращая код ошибки "0" (или иначе, но суть - Ok)





Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 1

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

HTML Слайда 1 (Бордюр)

HTML Лейера 2

вопрос-картинка

номер вопроса в туре

Менеджер Слайдов (VueSlideManager)

VueSlide

JS код компонента VueSlide

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

HTML Слайда 2 (Бордюр)

HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка

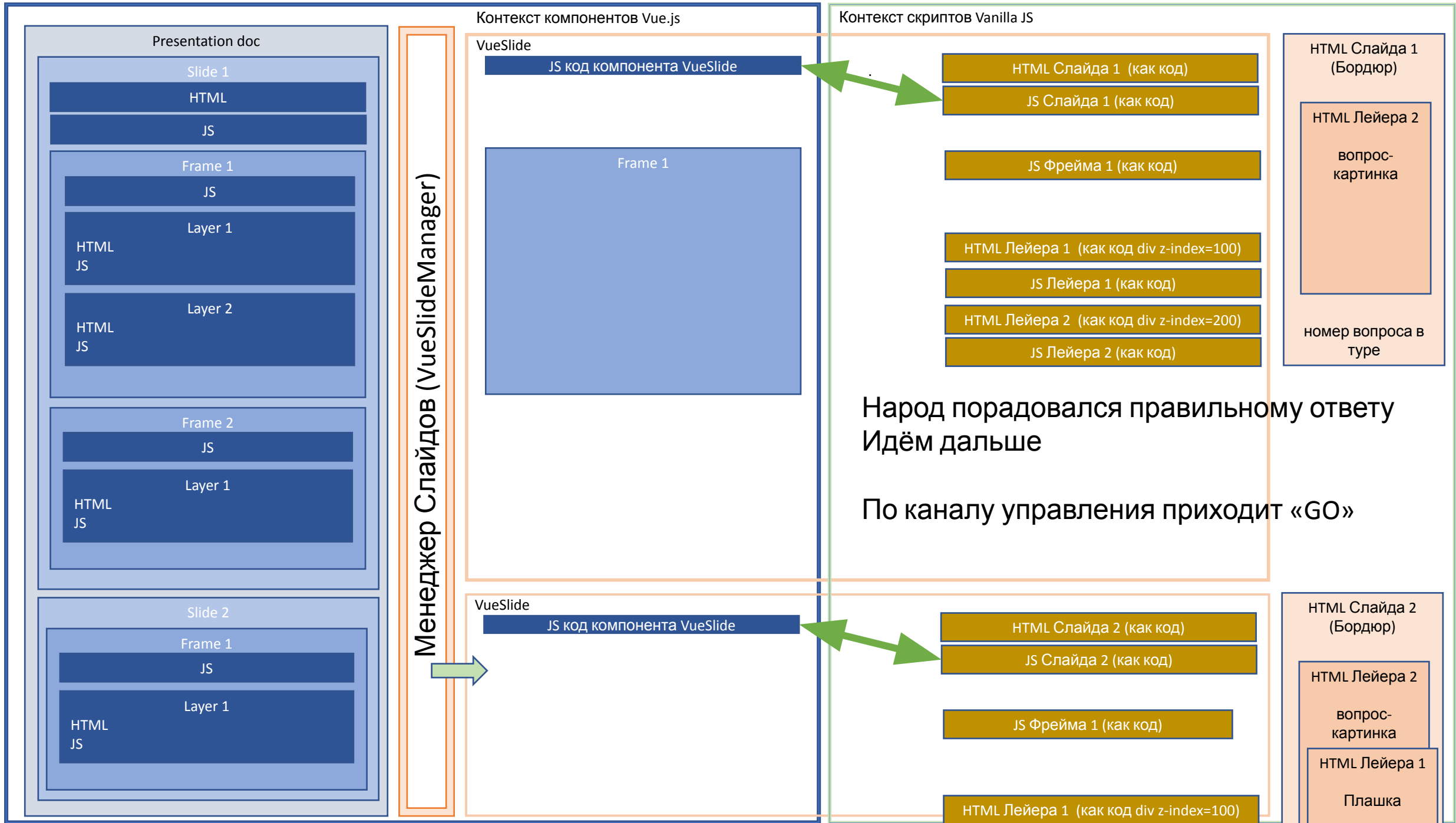
10.Ok

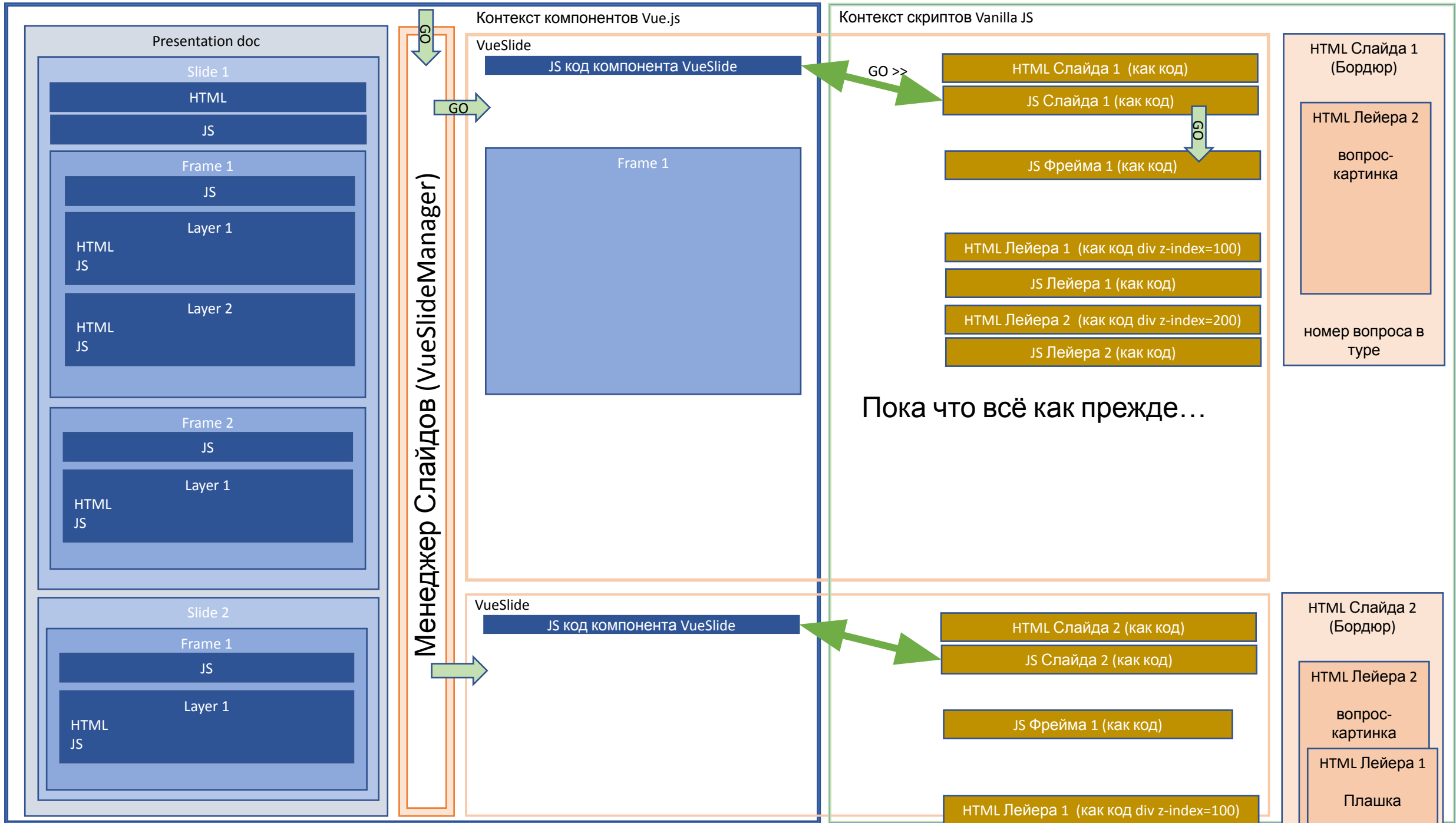
9.Ok

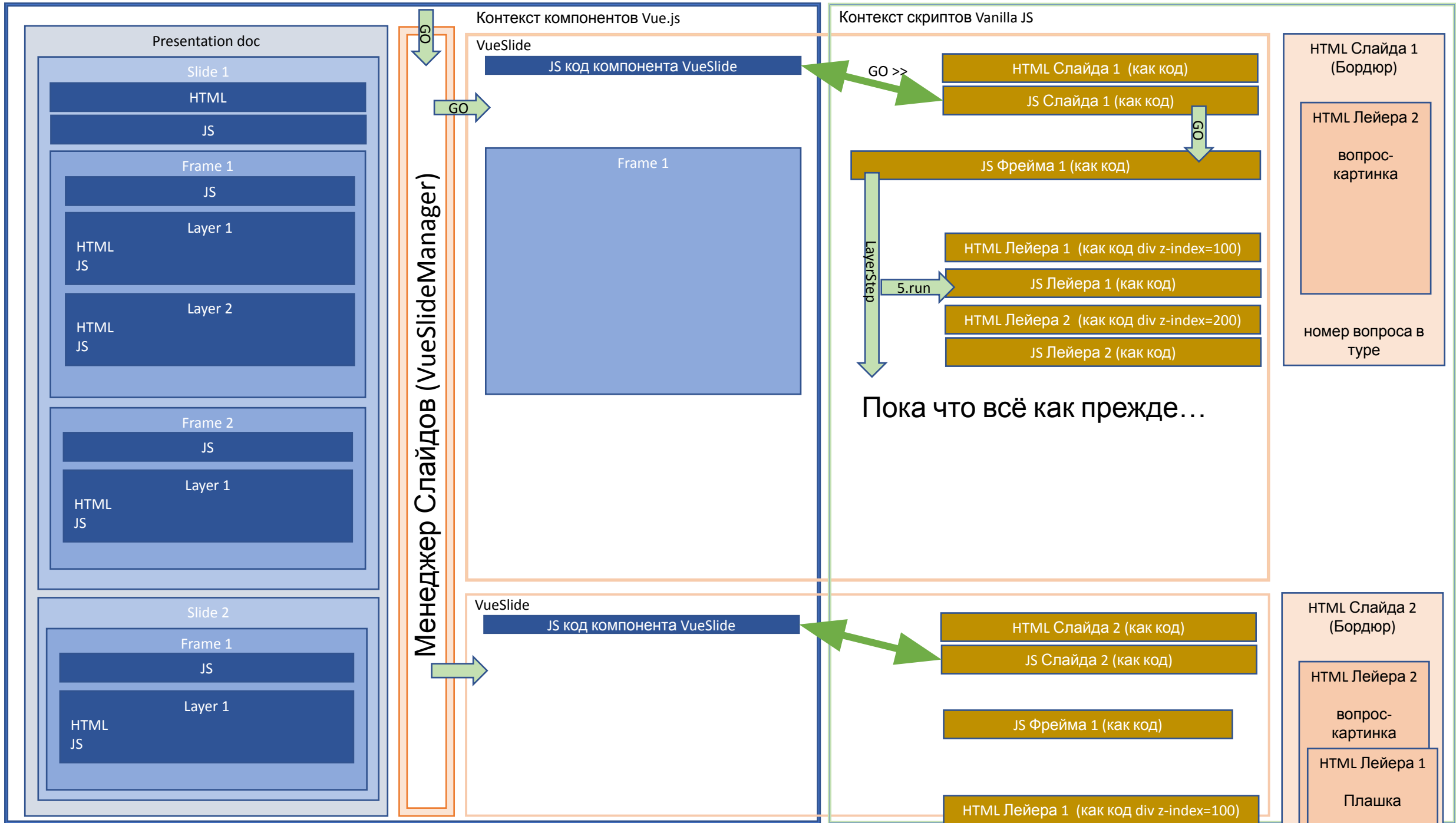
<< 8.Ok

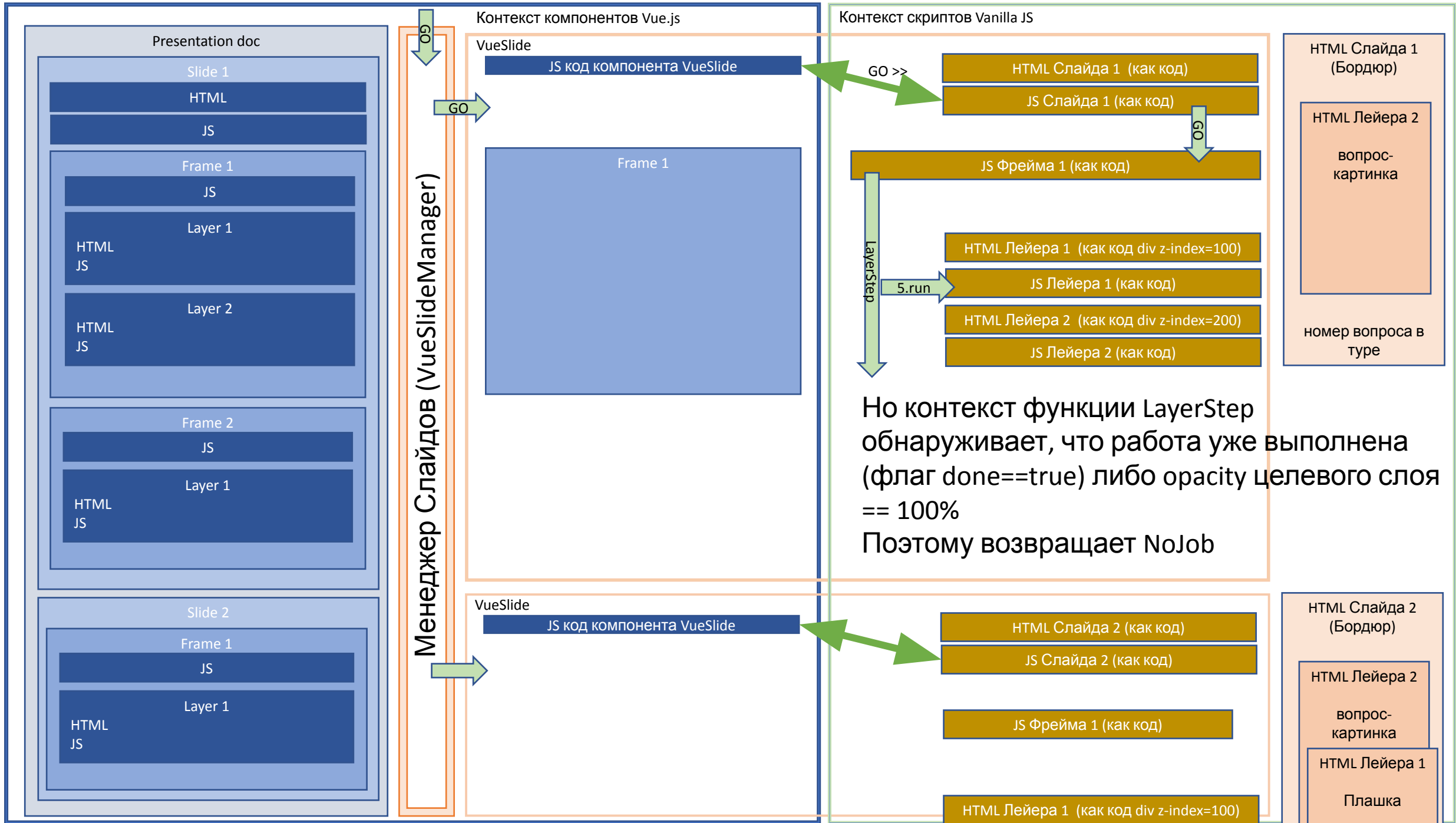
7.Ok

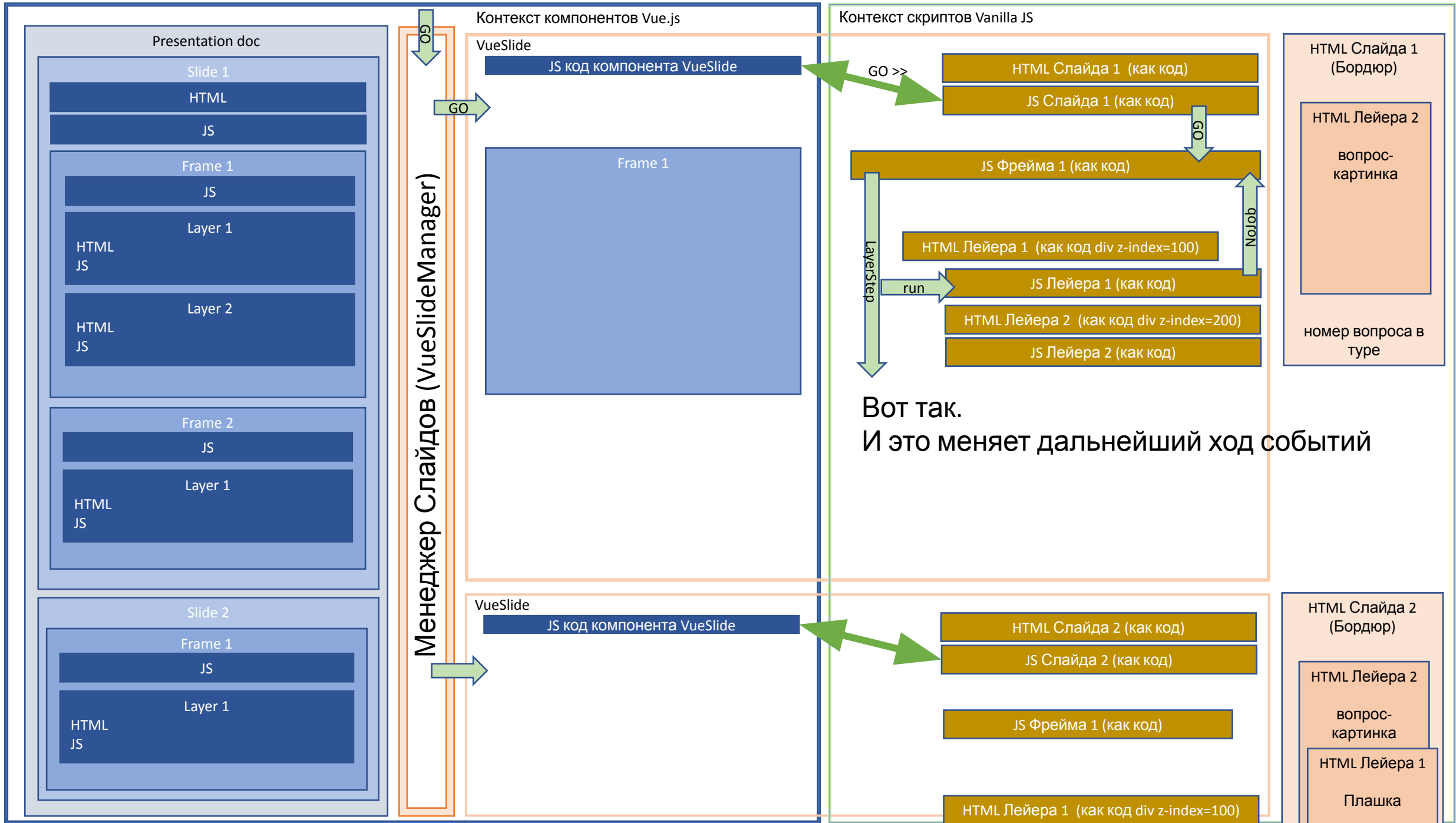
- 7. Вернули Ok
- 8. Вернули Ok
- 9. Вернули Ok
- 10. Вернули Ok

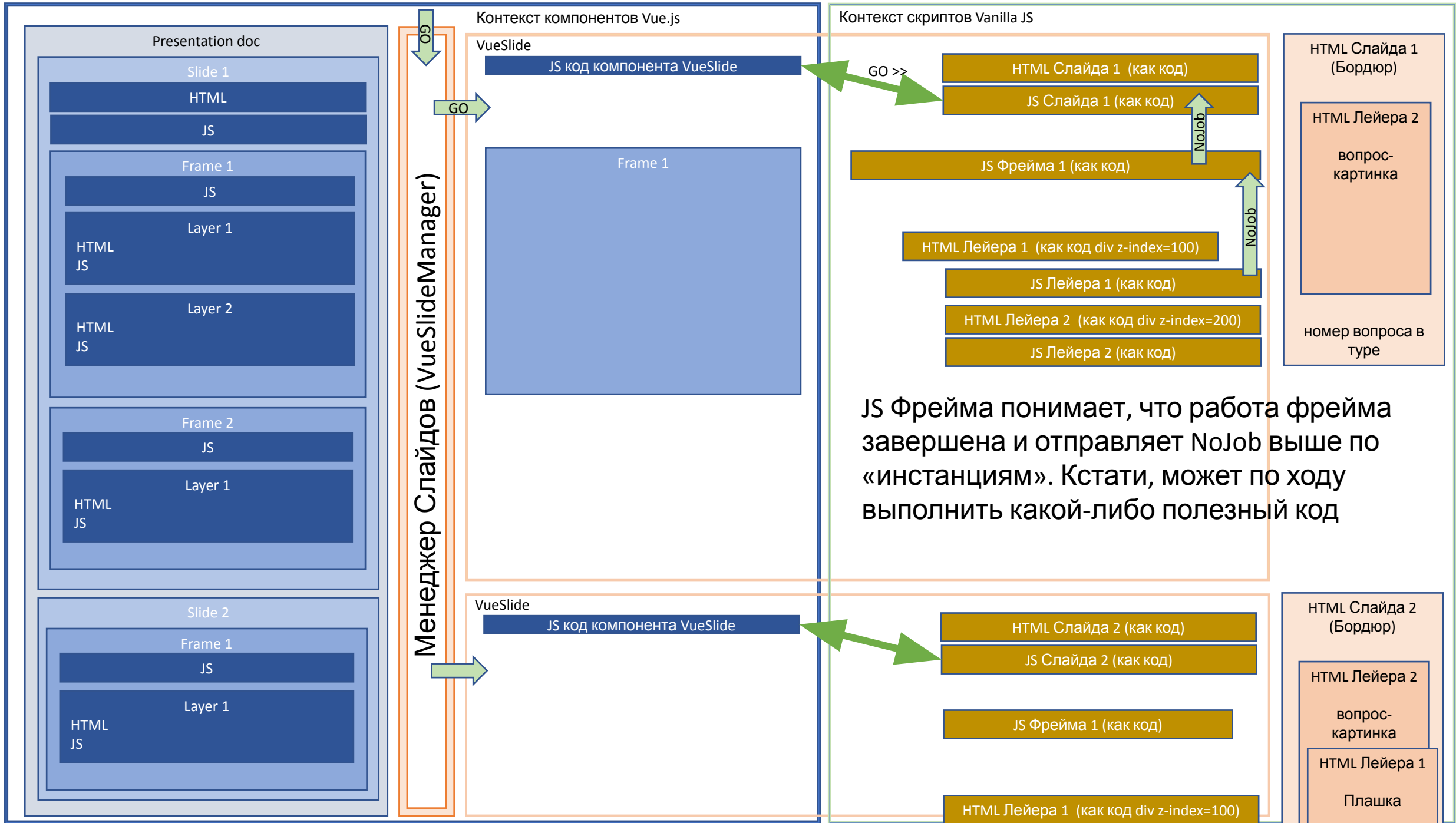




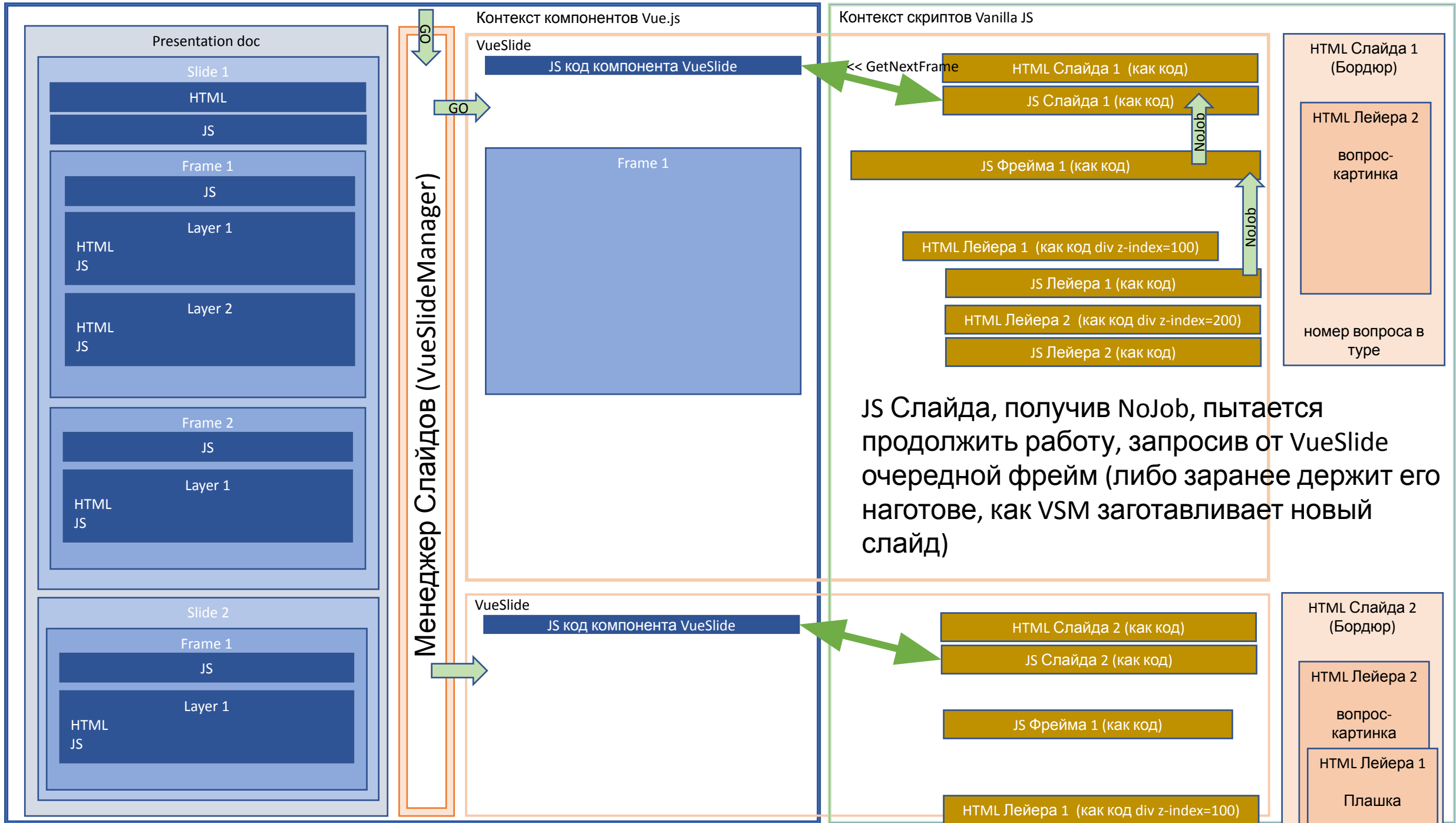








JS Фрейма понимает, что работа фрейма завершена и отправляет NoJob выше по «инстанциям». Кстати, может по ходу выполнить какой-либо полезный код



Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 1

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

<< GetNextFrame

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

JS Слайда, получив NoJob, пытается продолжить работу, запросив от VueSlide очередной фрейм (либо заранее держит его наготове, как VSM заготавливает новый слайд)

HTML Слайда 1 (Бордюр)

HTML Лейера 2

вопрос-картинка

номер вопроса в туре

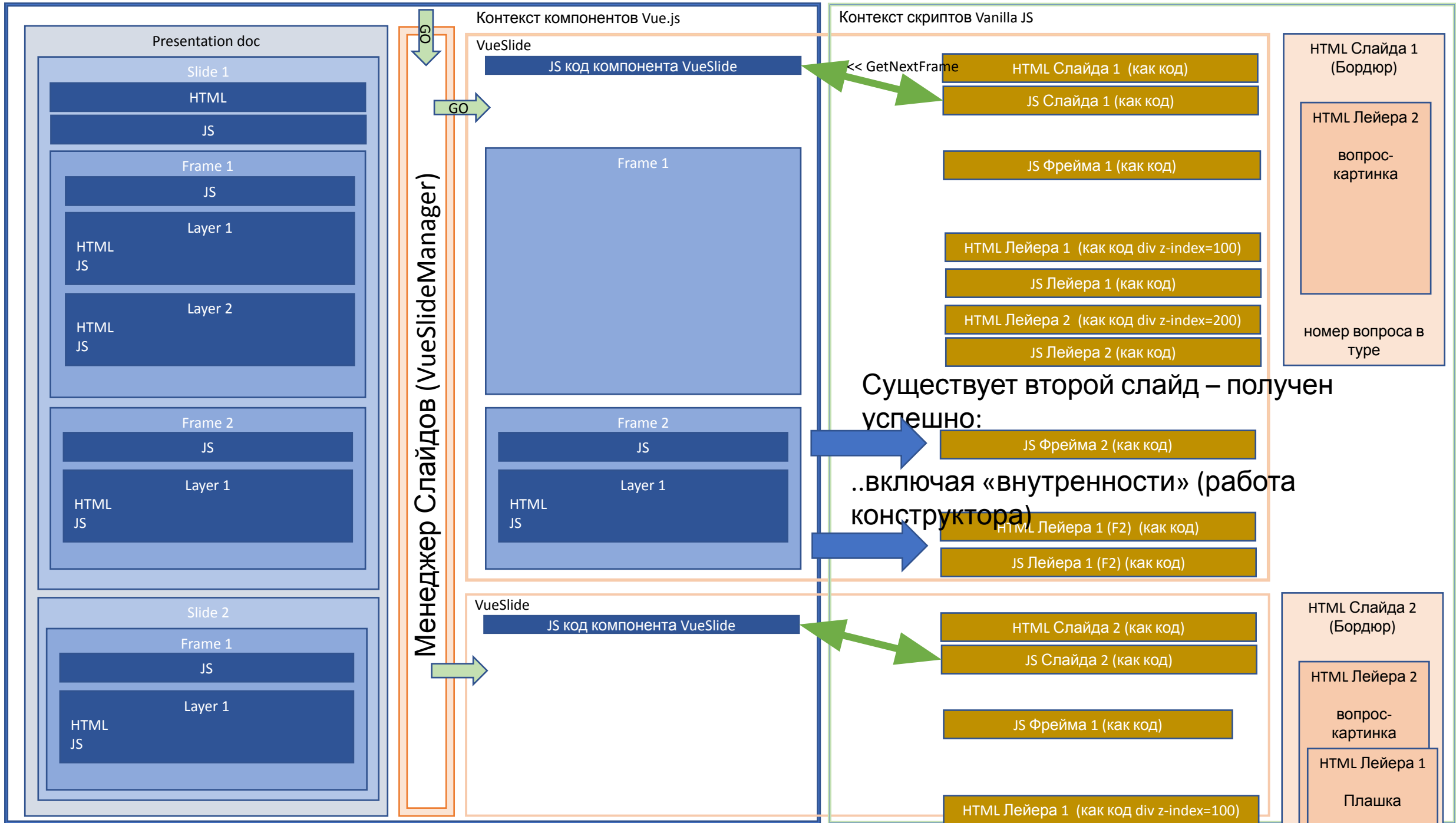
HTML Слайда 2 (Бордюр)

HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка



Presentation doc

- Slide 1
 - HTML
 - JS
 - Frame 1
 - JS
 - Layer 1
 - HTML JS
 - Layer 2
 - HTML JS
 - Frame 2
 - JS
 - Layer 1
 - HTML JS
- Slide 2
 - Frame 1
 - JS
 - Layer 1
 - HTML JS

Менеджер Слайдов (VueSlideManager)

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 1

Frame 2

JS

HTML JS

Layer 1

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

<< GetNextFrame

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

JS Фрейма 2 (как код)

HTML Лейера 1 (F2) (как код)

JS Лейера 1 (F2) (как код)

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

HTML Слайда 1 (Бордюр)

HTML Лейера 2

вопрос-картинка

номер вопроса в туре

HTML Слайда 2 (Бордюр)

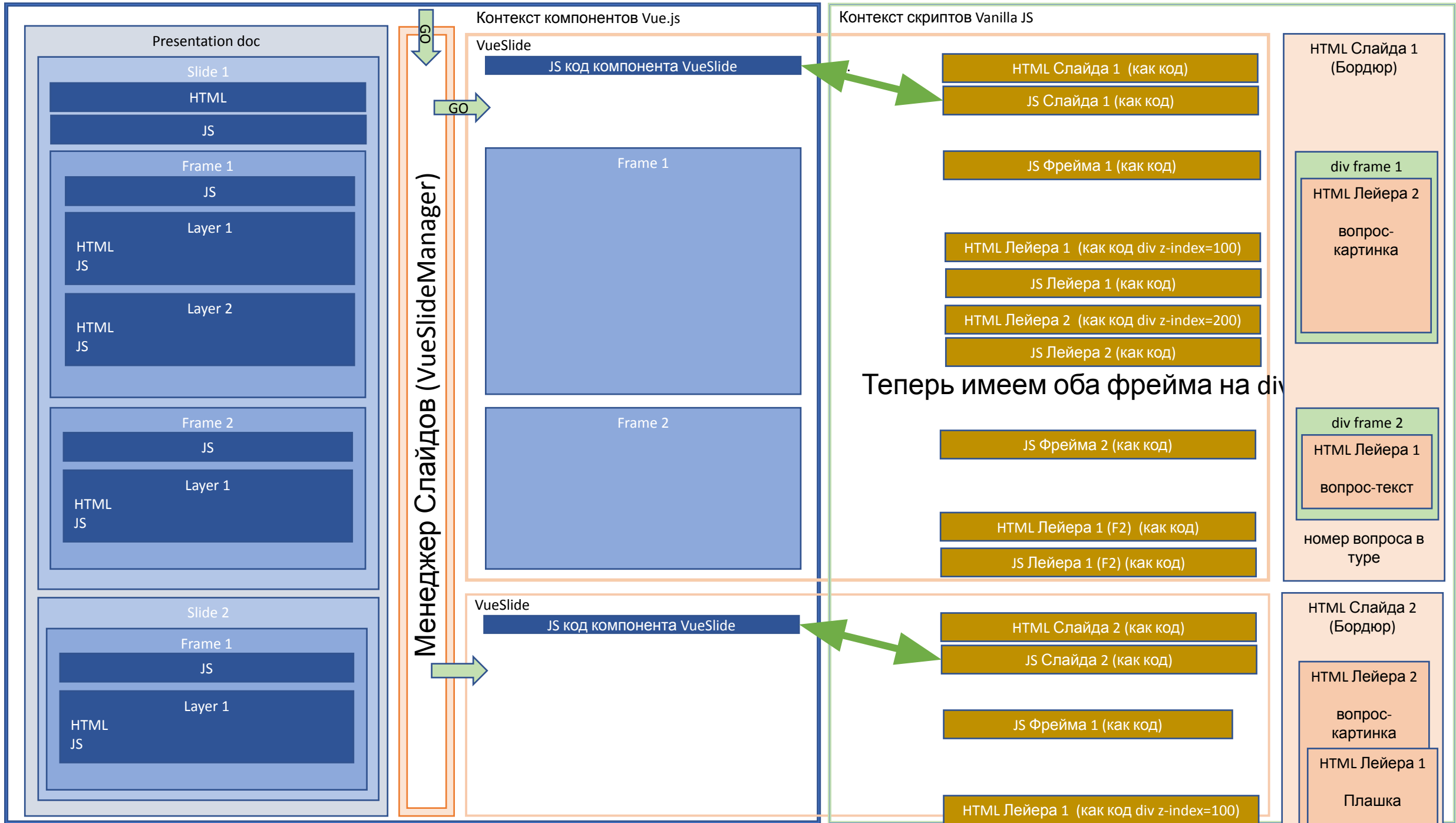
HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка

Существует второй слайд – получен успешно:
 ..включая «внутренности» (работа конструктора)



Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 1

Frame 2

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

JS Фрейма 2 (как код)

HTML Лейера 1 (F2) (как код)

JS Лейера 1 (F2) (как код)

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

HTML Слайда 1 (Бордюр)

div frame 1

HTML Лейера 2

вопрос-картинка

div frame 2

HTML Лейера 1

вопрос-текст

номер вопроса в туре

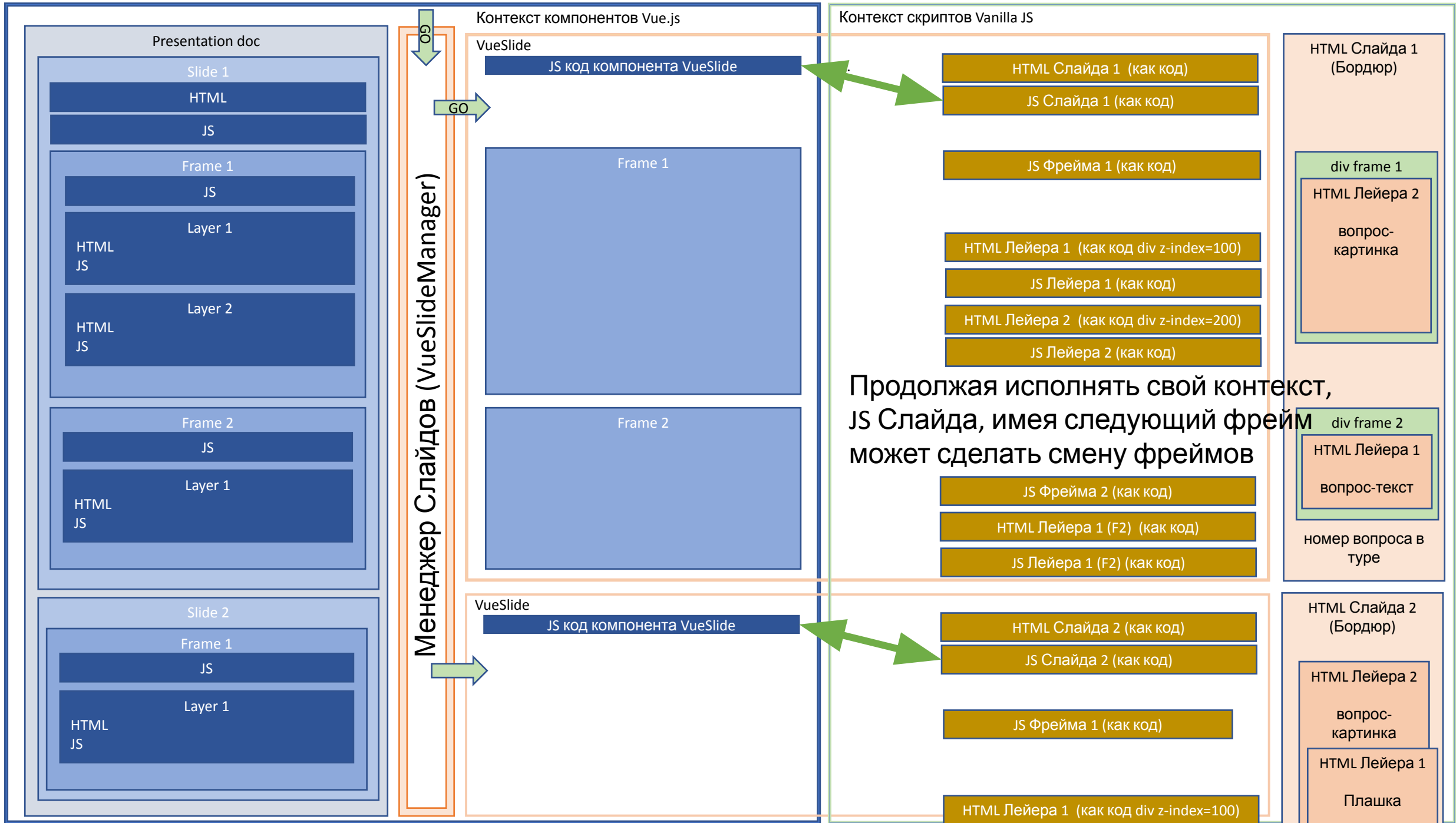
HTML Слайда 2 (Бордюр)

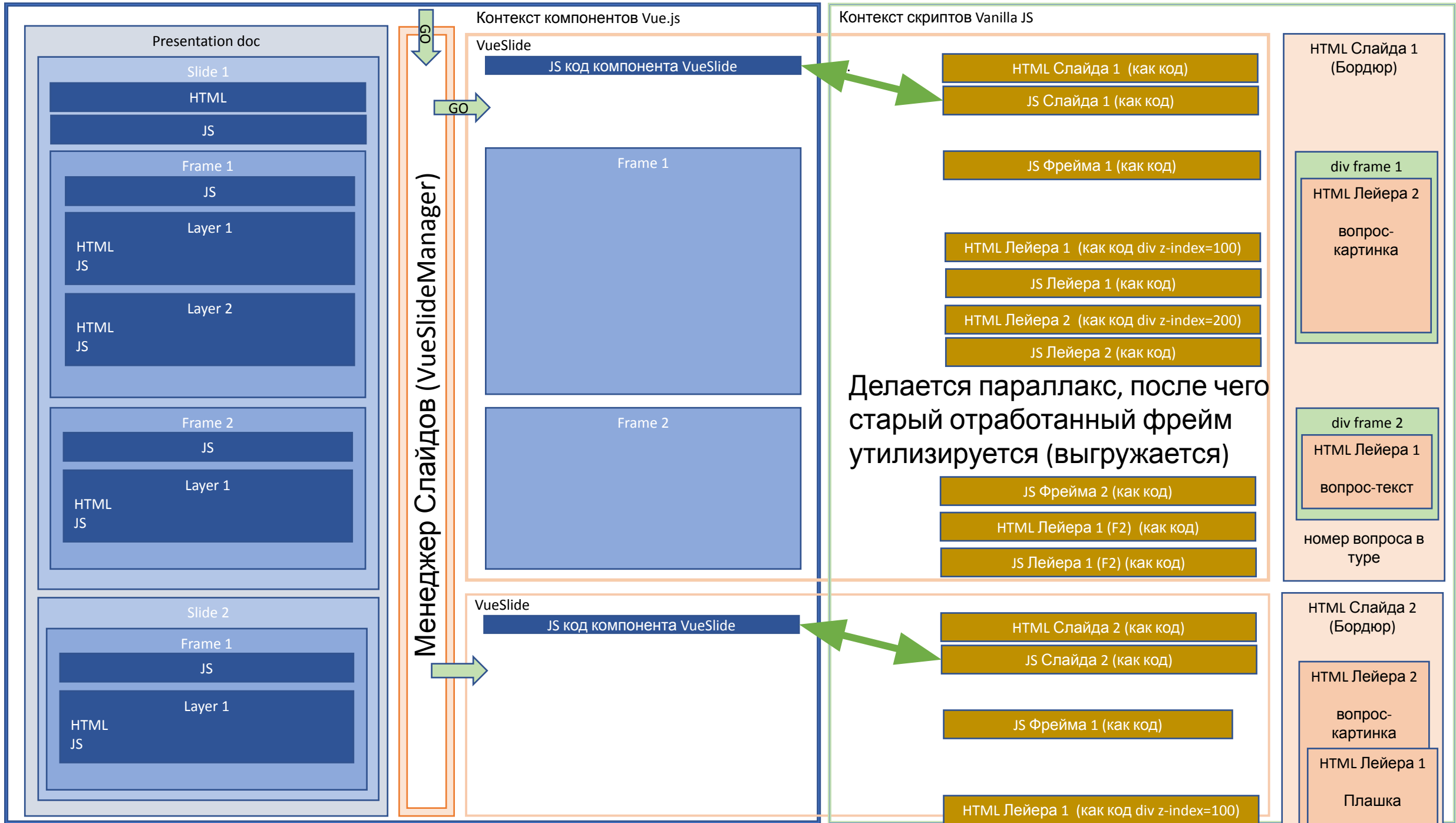
HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка





Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 1

Frame 2

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

JS Фрейма 2 (как код)

HTML Лейера 1 (F2) (как код)

JS Лейера 1 (F2) (как код)

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

Менеджер Слайдов (VueSlideManager)

GO

GO

Делается параллакс, после чего старый отработанный фрейм утилизируется (выгружается)

HTML Слайда 1 (Бордюр)

div frame 1

HTML Лейера 2

вопрос-картинка

div frame 2

HTML Лейера 1

вопрос-текст

номер вопроса в туре

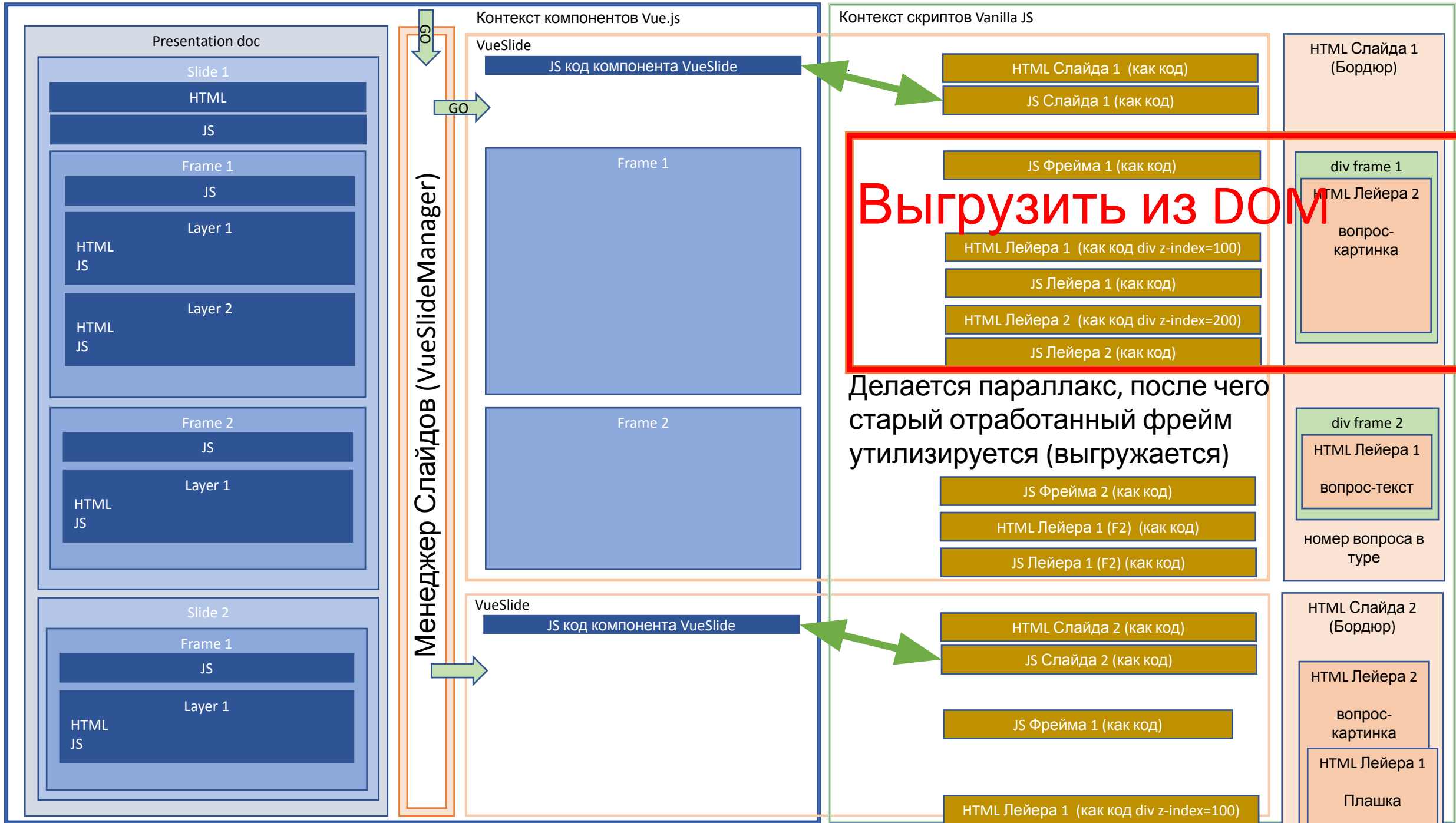
HTML Слайда 2 (Бордюр)

HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка



Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 1

Frame 2

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

JS Лейера 1 (как код)

HTML Лейера 2 (как код div z-index=200)

JS Лейера 2 (как код)

JS Фрейма 2 (как код)

HTML Лейера 1 (F2) (как код)

JS Лейера 1 (F2) (как код)

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

HTML Слайда 1 (Бордюр)

div frame 1

HTML Лейера 2

вопрос-картинка

div frame 2

HTML Лейера 1

вопрос-текст

номер вопроса в туре

HTML Слайда 2 (Бордюр)

HTML Лейера 2

вопрос-картинка

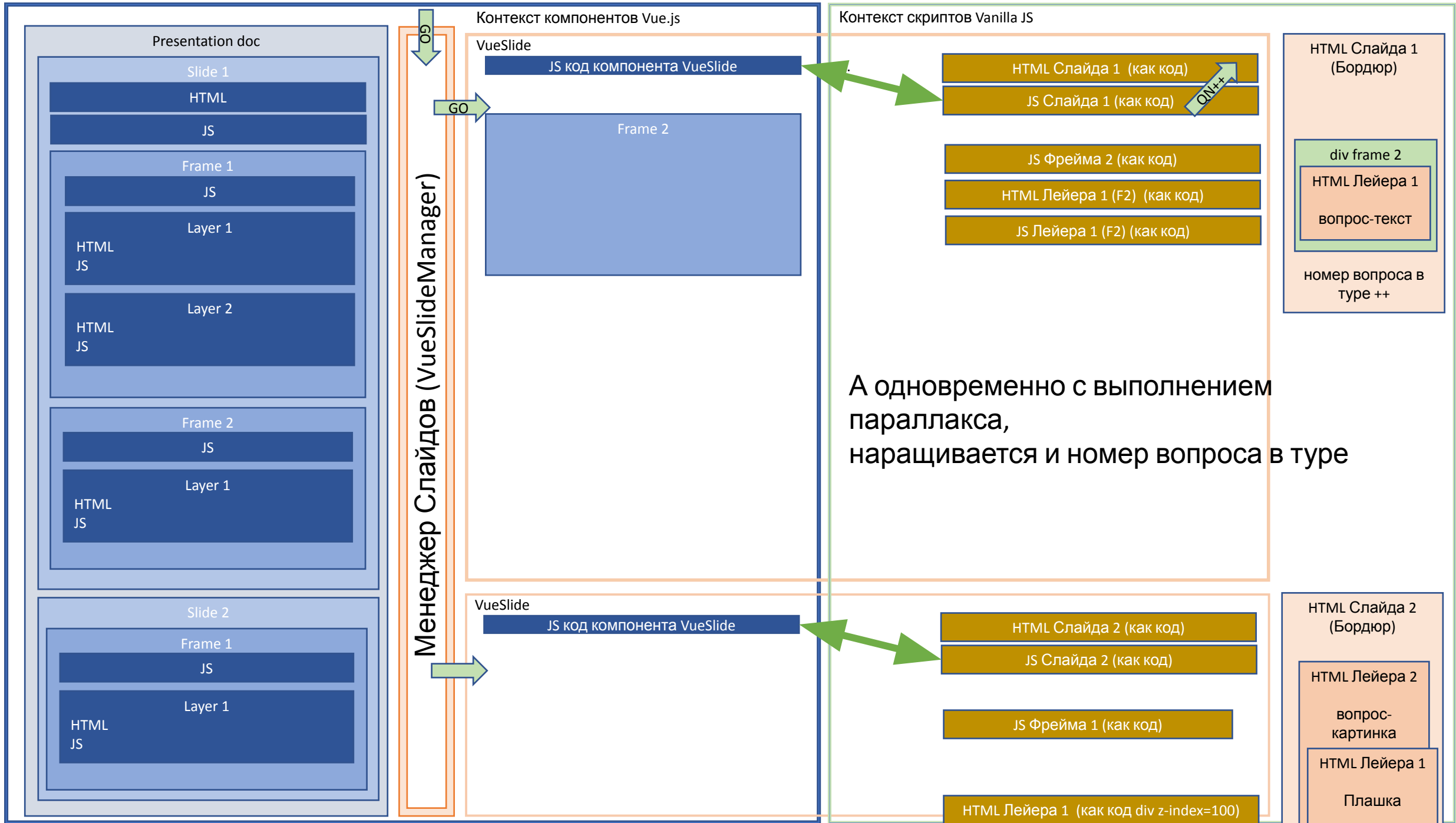
HTML Лейера 1

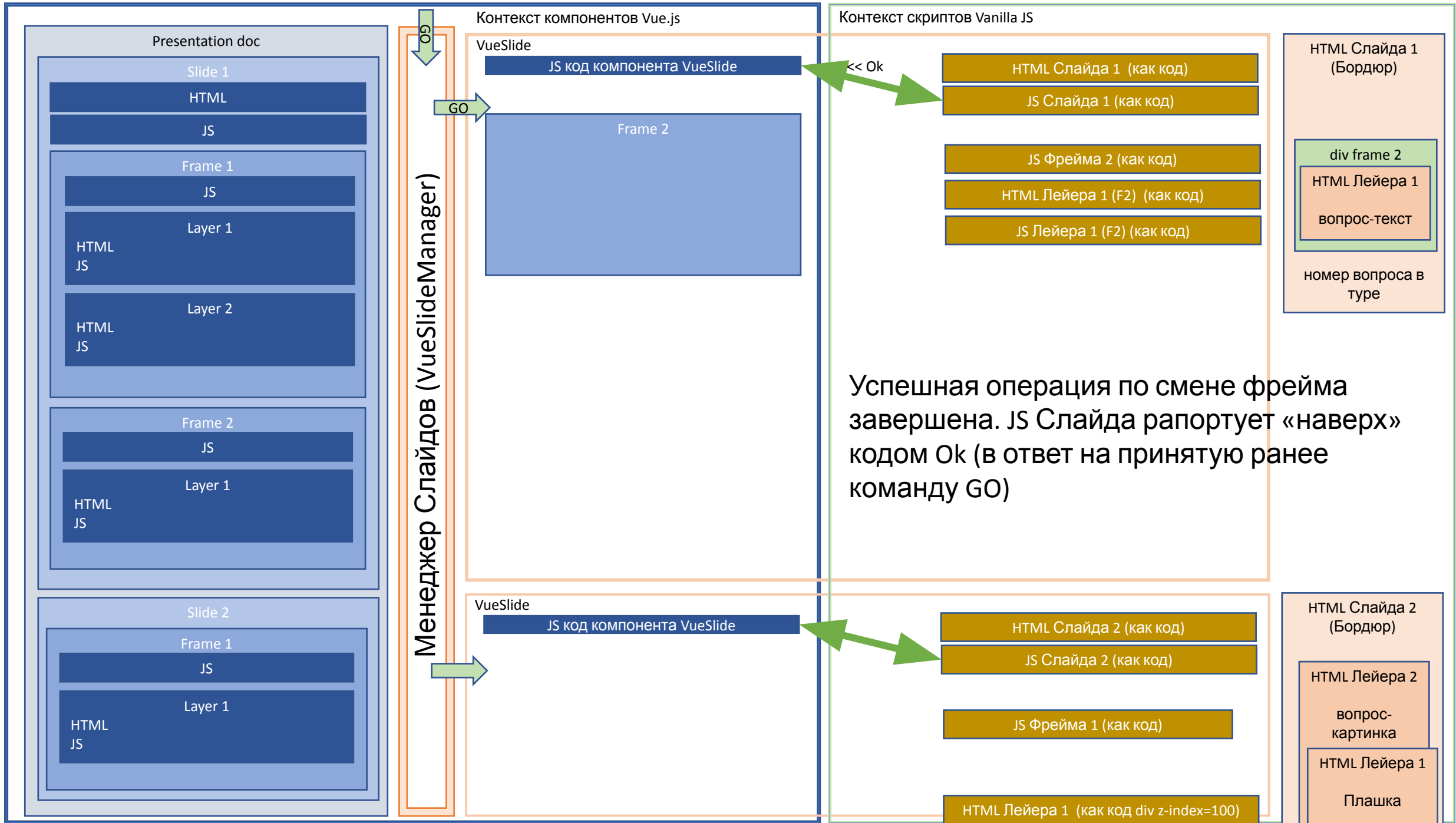
Плашка

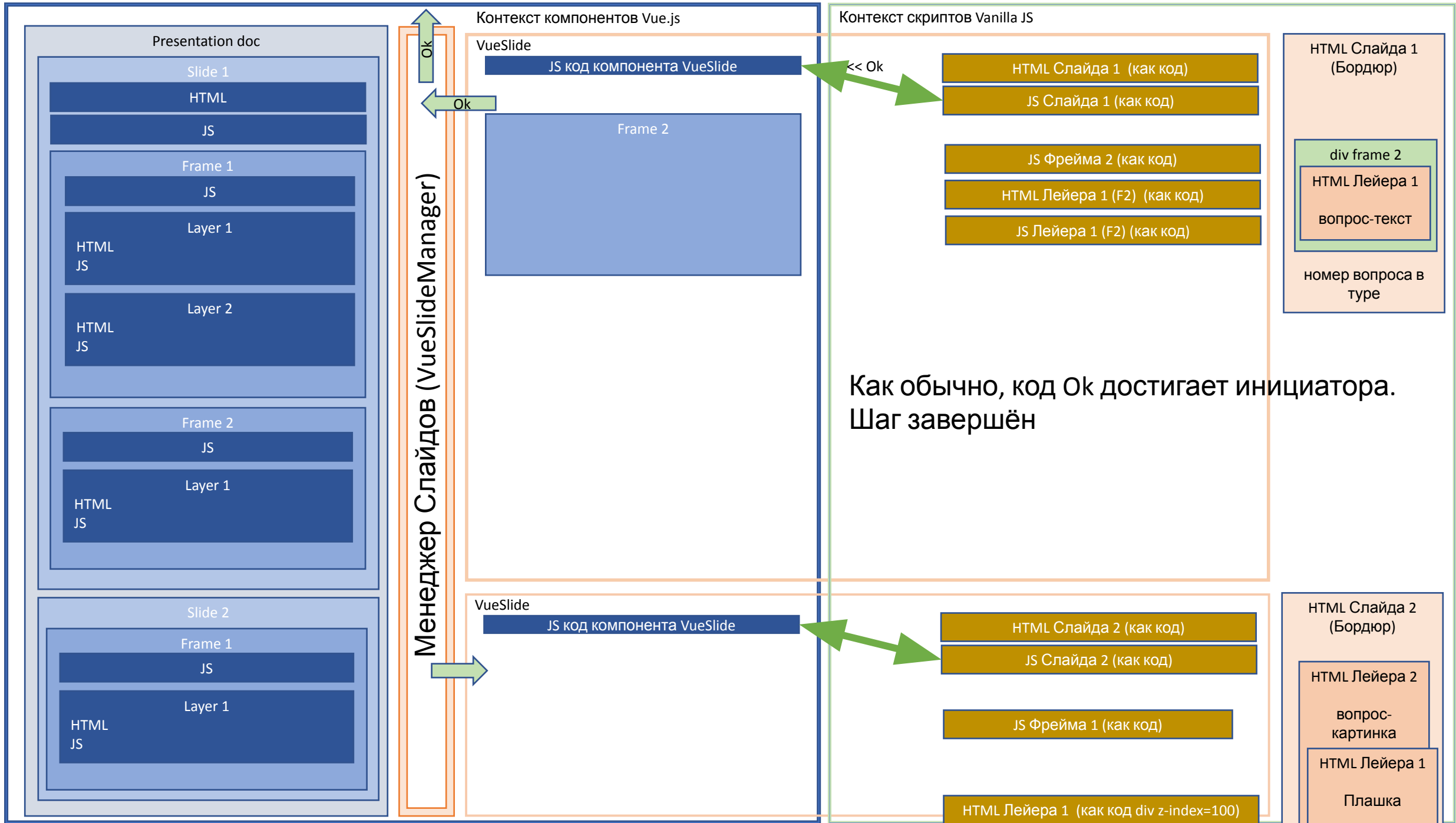
Менеджер Слайдов (VueSlideManager)

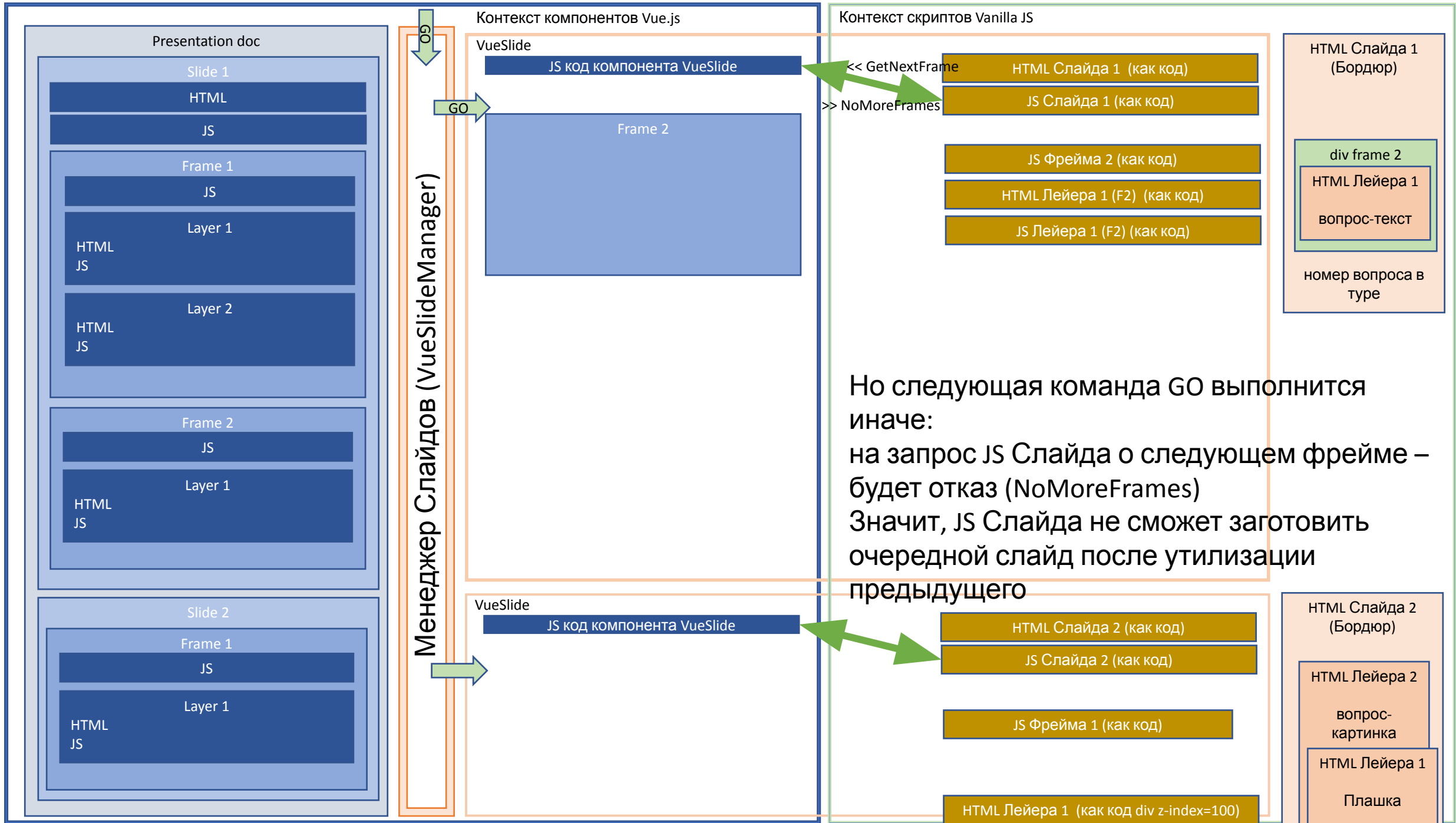
Выгрузить из DOM

Делается параллакс, после чего старый отработанный фрейм утилизируется (выгружается)









Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 2

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 2 (как код)

HTML Лейера 1 (F2) (как код)

JS Лейера 1 (F2) (как код)

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

HTML Слайда 1 (Бордюр)

div frame 2

HTML Лейера 1

вопрос-текст

номер вопроса в
туре

HTML Слайда 2 (Бордюр)

HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка

<< GetNextFrame

>> NoMoreFrames

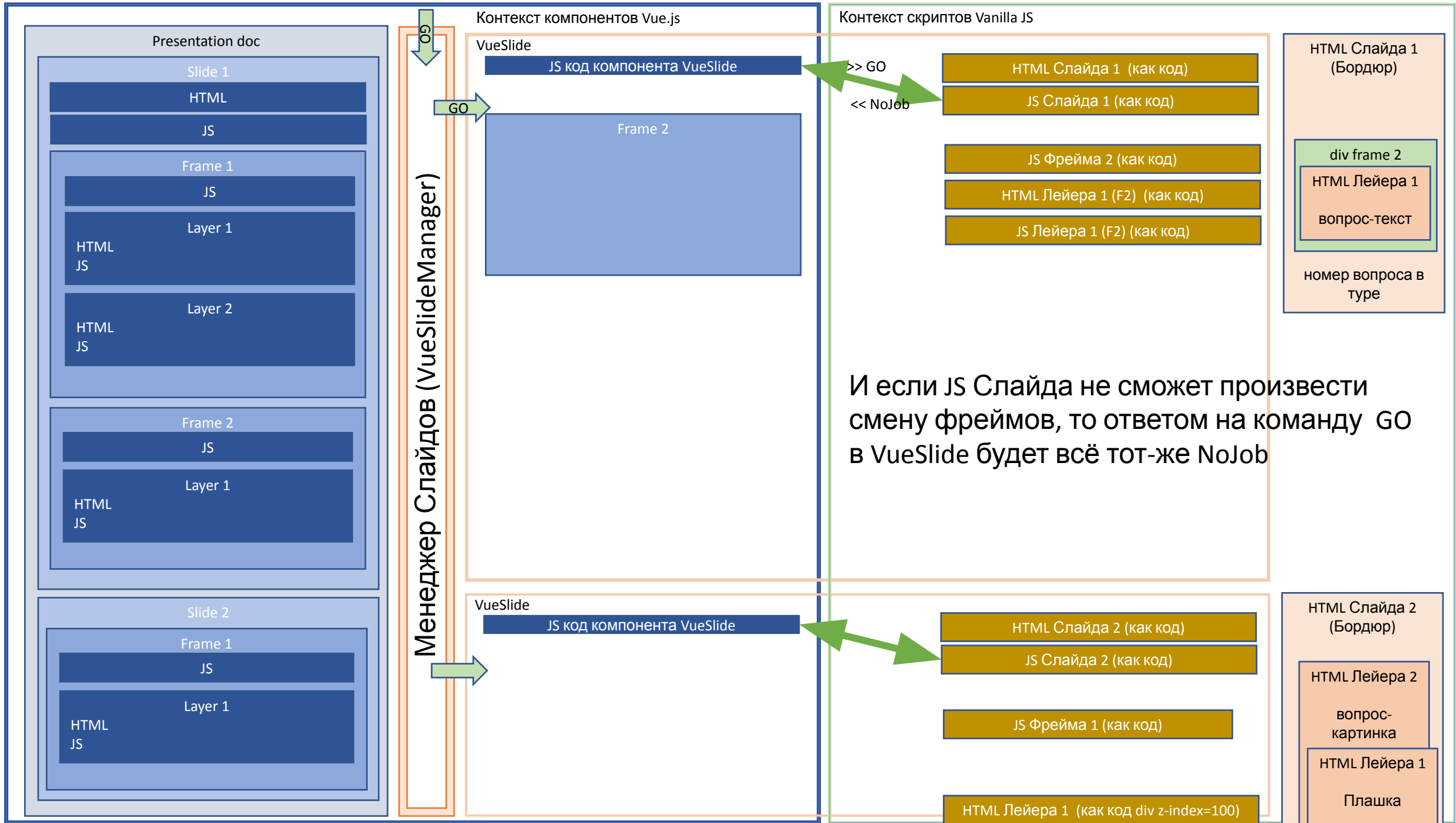
Менеджер Слайдов (VueSlideManager)

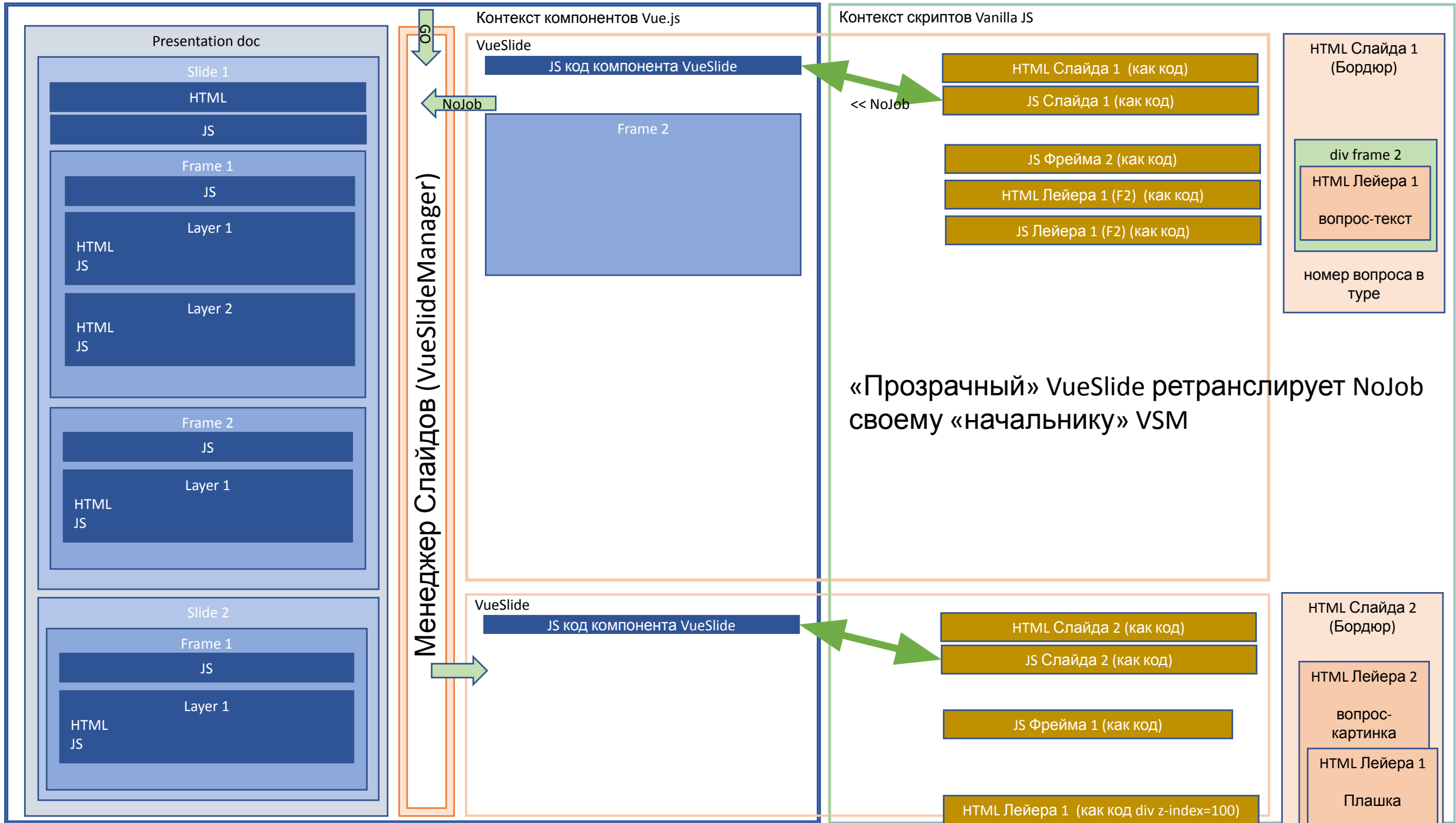
GO

GO

<< GetNextFrame

>> NoMoreFrames





Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

Frame 2

VueSlide

JS код компонента VueSlide

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 2 (как код)

HTML Лейера 1 (F2) (как код)

JS Лейера 1 (F2) (как код)

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

Менеджер Слайдов (VueSlideManager)

NoJob

<< NoJob

«Прозрачный» VueSlide ретранслирует NoJob своему «начальнику» VSM

HTML Слайда 1 (Бордюр)

div frame 2

HTML Лейера 1

вопрос-текст

номер вопроса в
туре

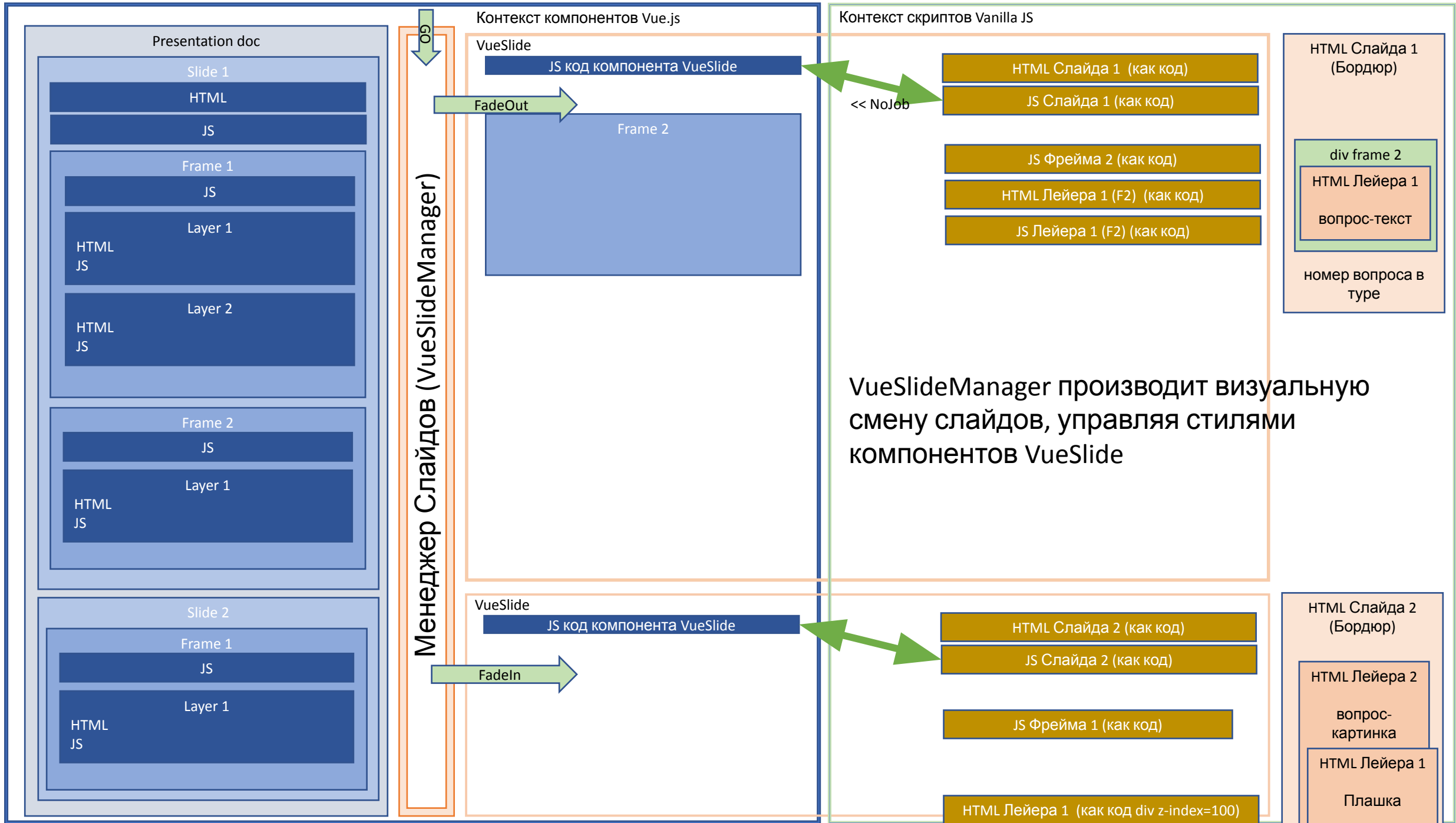
HTML Слайда 2 (Бордюр)

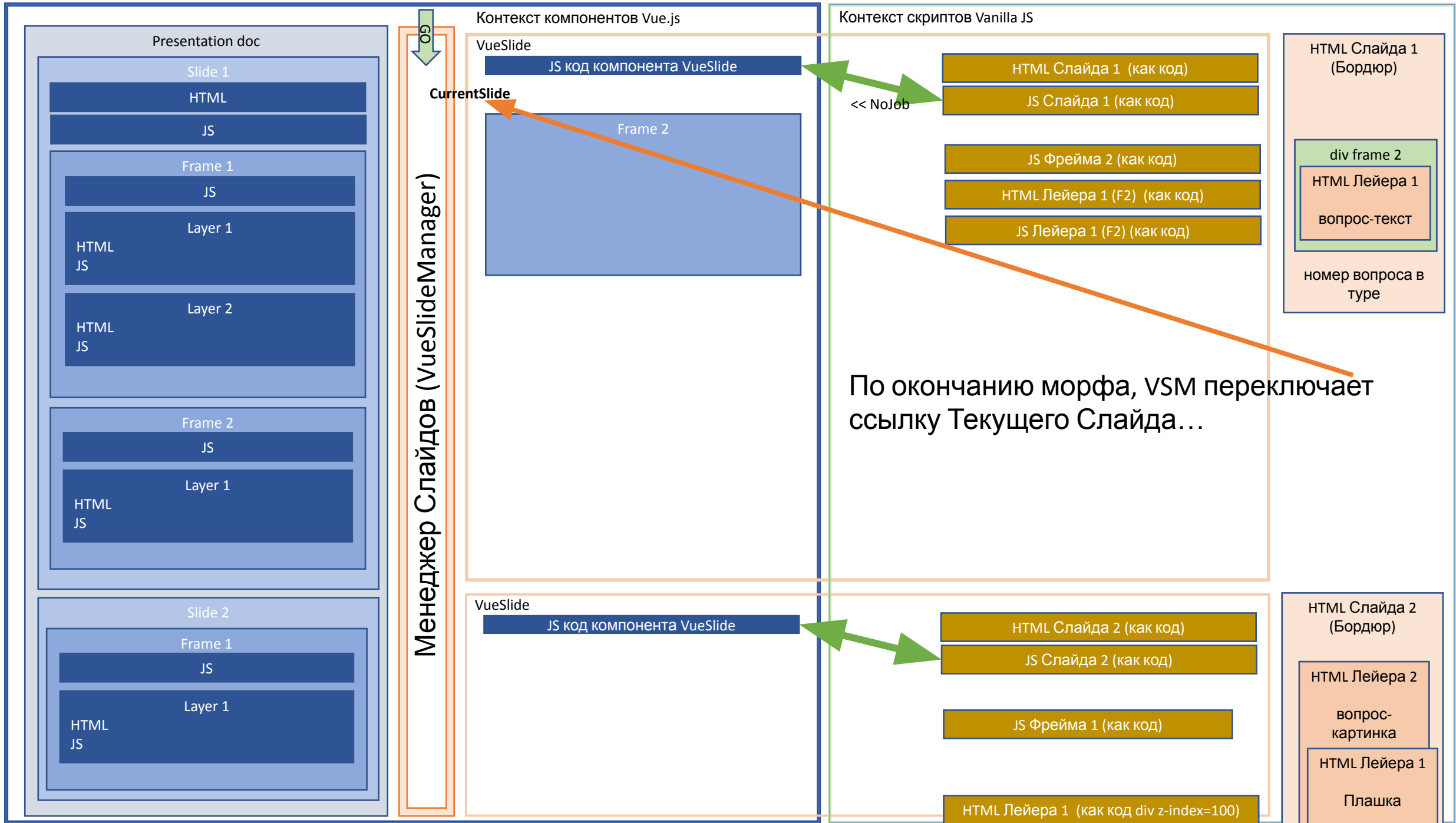
HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка





Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Контекст компонентов Vue.js

VueSlide

JS код компонента VueSlide

CurrentSlide

Frame 2

Контекст скриптов Vanilla JS

HTML Слайда 1 (как код)

JS Слайда 1 (как код)

JS Фрейма 2 (как код)

HTML Лейера 1 (F2) (как код)

JS Лейера 1 (F2) (как код)

HTML Слайда 1
(Бордюр)

div frame 2

HTML Лейера 1

вопрос-текст

номер вопроса в
туре

Менеджер Слайдов (VueSlideManager)

<< NoJob

По окончании морфа, VSM переключает
ссылку Текущего Слайда...

VueSlide

JS код компонента VueSlide

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

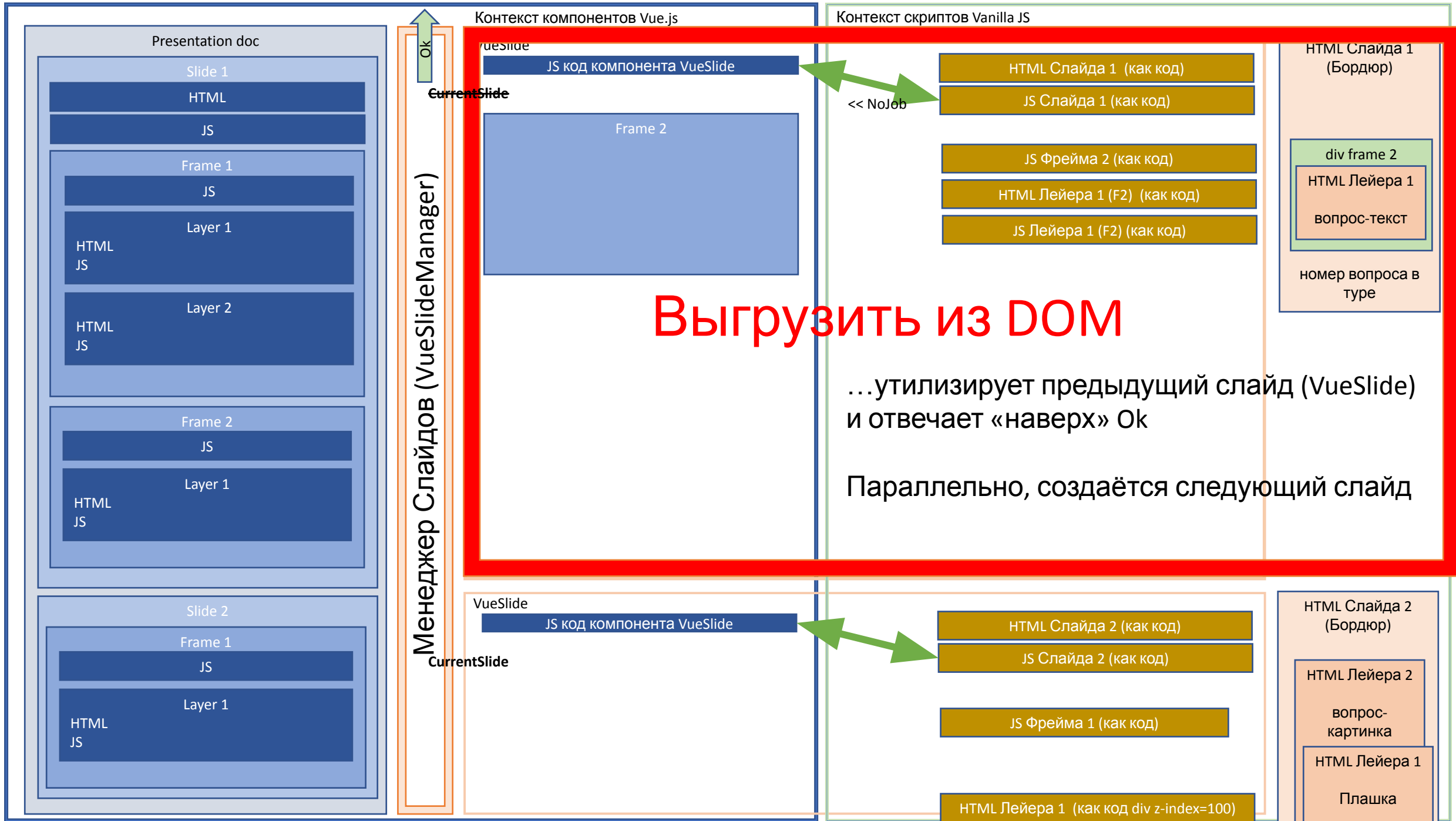
HTML Слайда 2
(Бордюр)

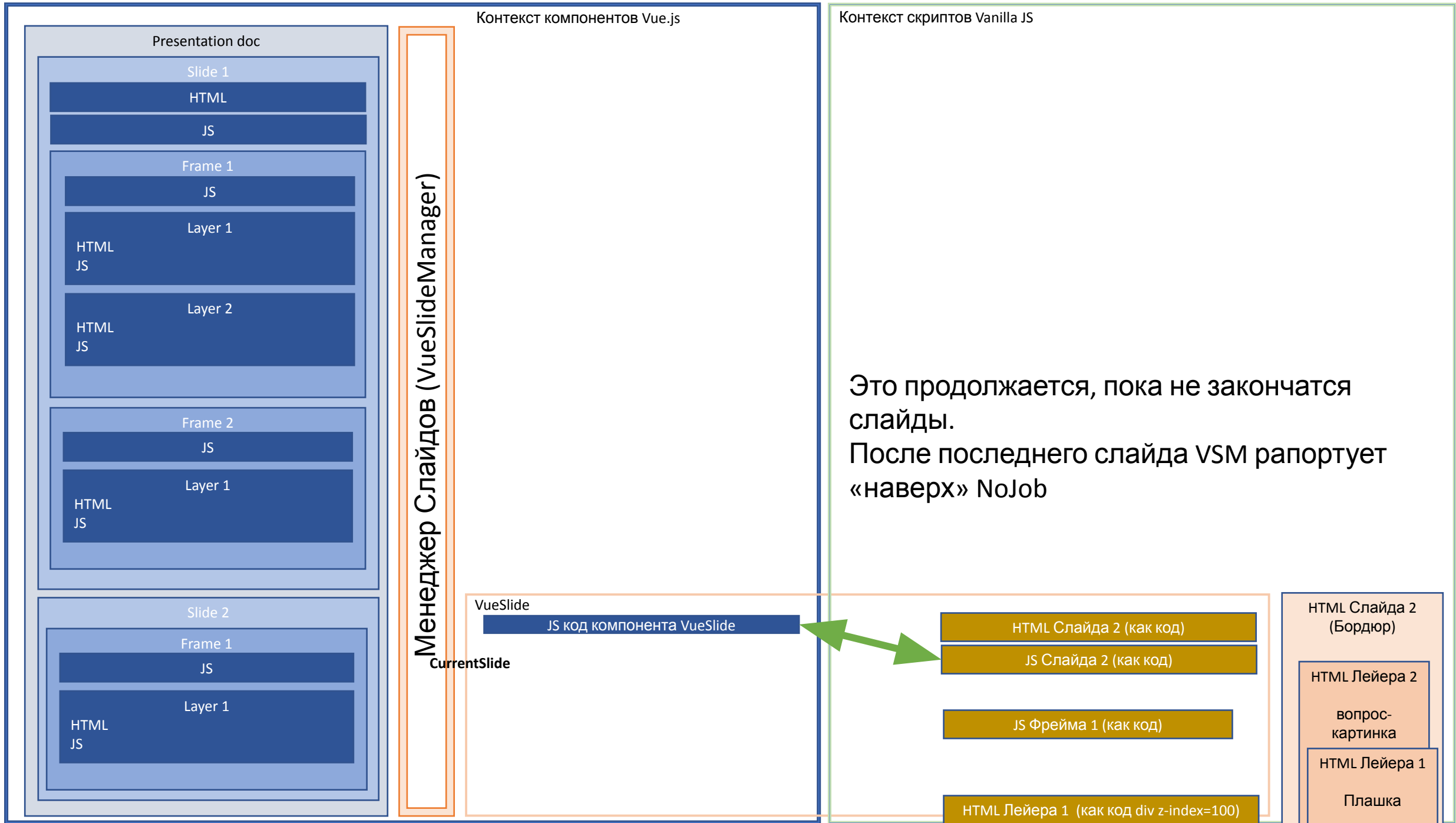
HTML Лейера 2

вопрос-
картинка

HTML Лейера 1

Плашка





Presentation doc

Slide 1

HTML

JS

Frame 1

JS

Layer 1

HTML
JS

Layer 2

HTML
JS

Frame 2

JS

Layer 1

HTML
JS

Slide 2

Frame 1

JS

Layer 1

HTML
JS

Менеджер Слайдов (VueSlideManager)

Контекст компонентов Vue.js

Контекст скриптов Vanilla JS

VueSlide

JS код компонента VueSlide

CurrentSlide

HTML Слайда 2 (как код)

JS Слайда 2 (как код)

JS Фрейма 1 (как код)

HTML Лейера 1 (как код div z-index=100)

HTML Слайда 2 (Бордюр)

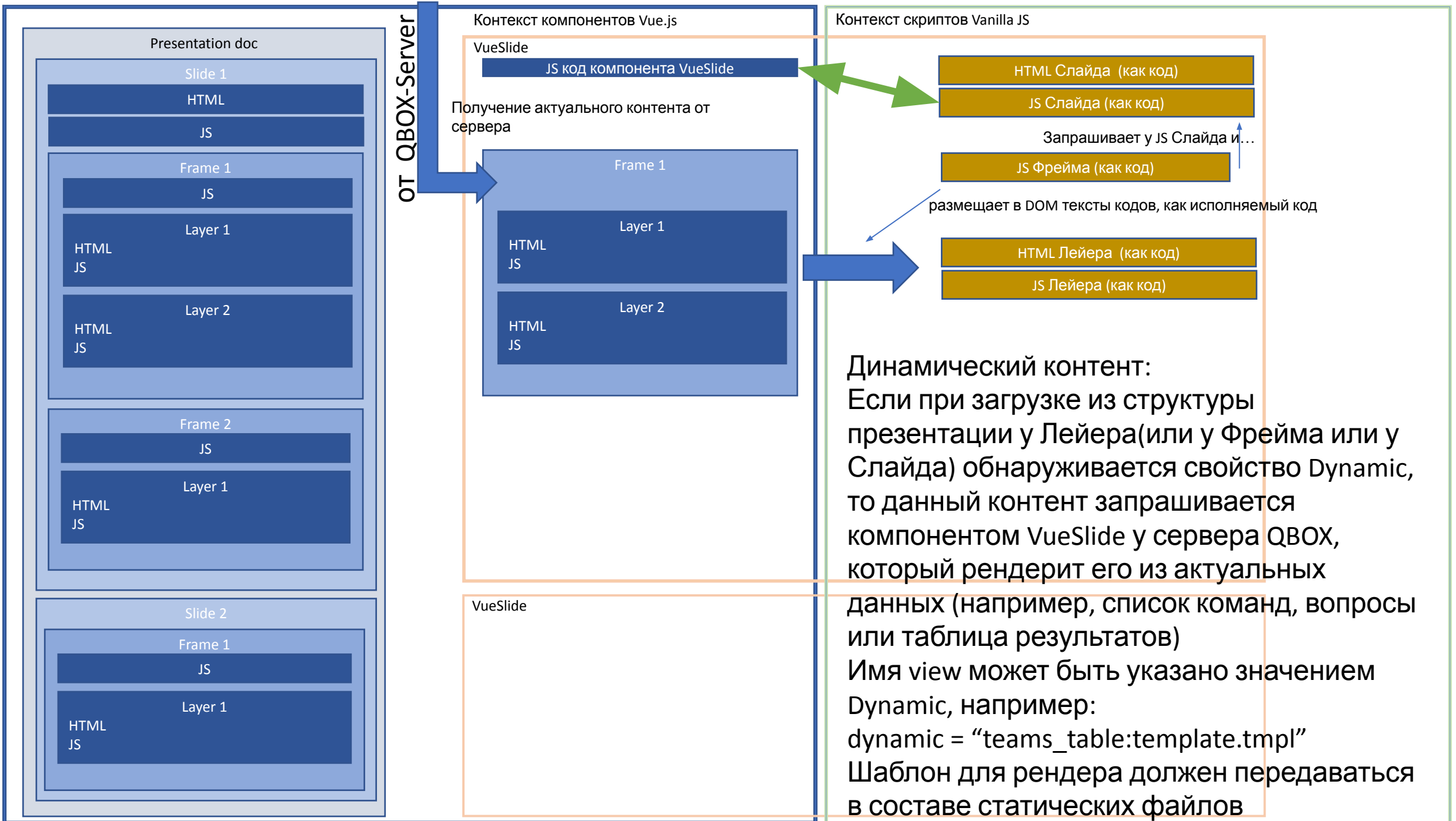
HTML Лейера 2

вопрос-картинка

HTML Лейера 1

Плашка

Это продолжается, пока не закончатся слайды.
После последнего слайда VSM рапортует «наверх» NoJob



Vcë!