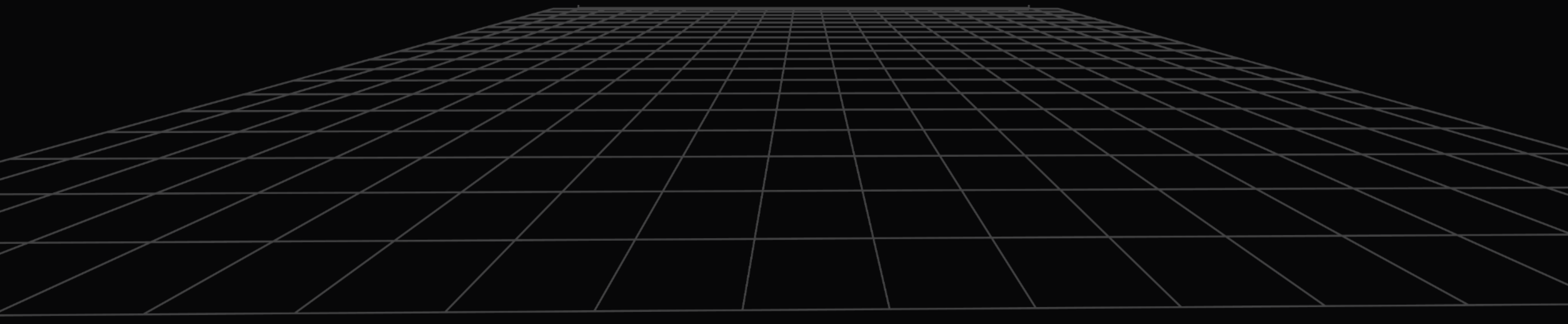


КОНТЕКСТЫ

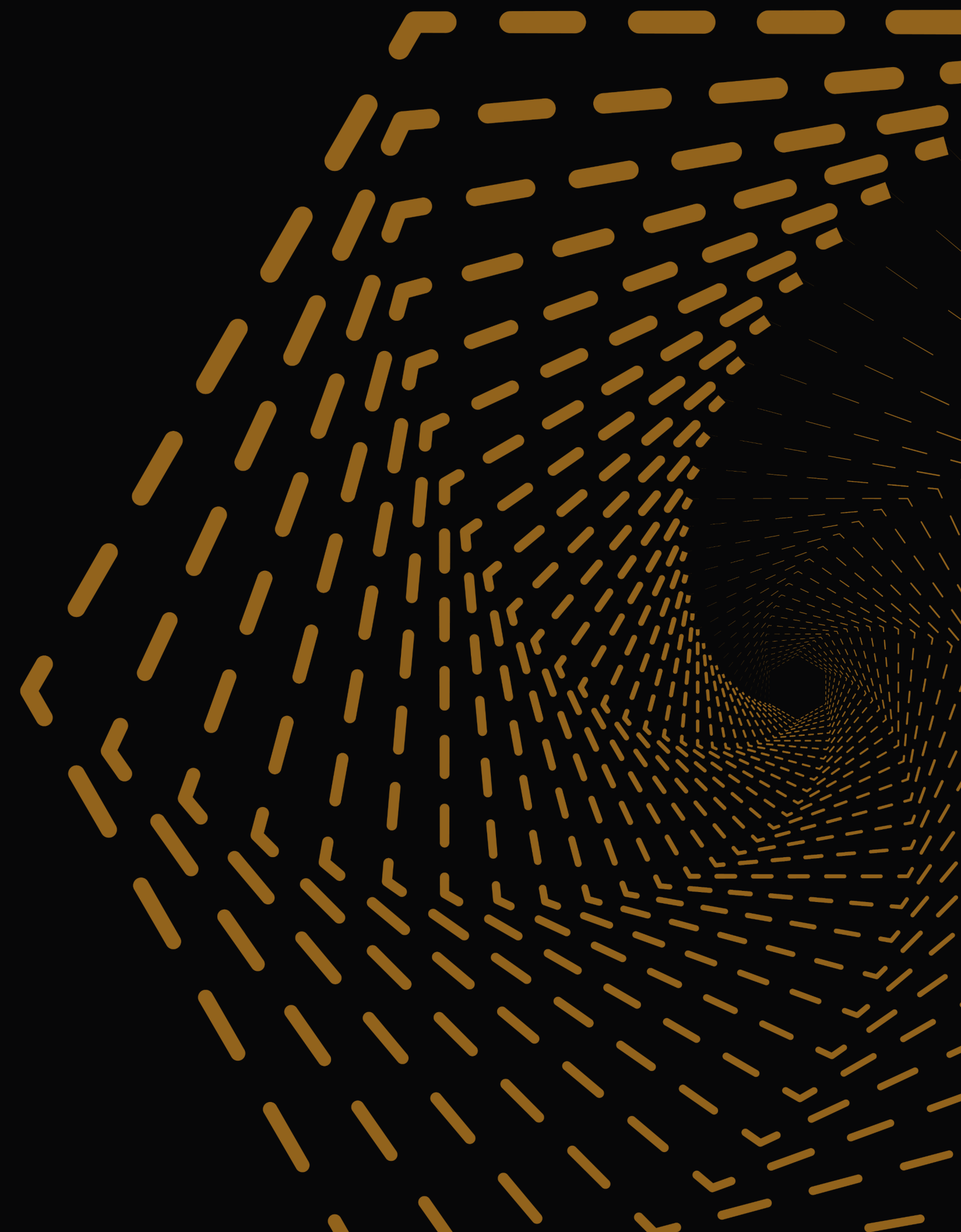


ПРОВЕРЬ ЗАПИСЬ



ПРАВИЛА ЗАНЯТИЯ

1. вопросы в чате можно задавать
в любое время
2. вопросы голосом задаем
по поднятой руке в Zoom
3. ответы на вопросы будут
в запланированных местах



Terminal: Concurrency × + ∨



КОНТЕКСТ – ЭТО...

Набор мета-данных, ассоциированных
с каким-либо запросом или процессом

КОНТЕКСТЫ



```
1 func Background() Context
2 func TODO() Context
3 func WithValue(parent Context, key, val any) Context
4 func WithoutCancel(parent Context) Context
5 func WithCancel(parent Context) (ctx Context, cancel CancelFunc)
6 func WithCancelCause(parent Context) (ctx Context, cancel CancelCauseFunc)
7 func WithDeadline(parent Context, d time.Time) (Context, CancelFunc)
8 func WithDeadlineCause(parent Context, d time.Time, cause error) (Context, CancelFunc)
9 func WithTimeout(parent Context, timeout time.Duration) (Context, CancelFunc)
10 func WithTimeoutCause(parent Context, timeout time.Duration, cause error) (Context, CancelFunc)
```

API



```
1 type Context struct { ... }  
2  
3 func (c *Context) Deadline() (time.Time, bool)           // возвращает время дедлайна и был ли он установлен  
4 func (c *Context) Done() <-chan struct{}                // возвращает канал, сигнализирующий об отмене  
5 func (c *Context) Err() error                           // возвращает ошибку, сигнализирующую об отмене  
6 func (c *Context) Value(key interface{}) interface{}    // возвращает установленное значение по ключу
```


**ЧЕМ ОТЛИЧАЕТСЯ
CONTEXT.WITHTIMEOUT
ОТ CONTEXT.WITHDEADLINE?**



```
1 func WithTimeout(parent Context, timeout time.Duration) (Context, CancelFunc) {  
2     return WithDeadline(parent, time.Now().Add(timeout))  
3 }
```


EXAMPLE

Context with cancel

Вызвать функцию отмены в другом месте
в Go считается **антипаттерном**

EXAMPLE

Check cancel

EXAMPLE

With ctx check

EXAMPLE

Context with timeout

Если **WithCancel** или **WithTimeout** не будет отменен
– в результате получится утечка горютины!

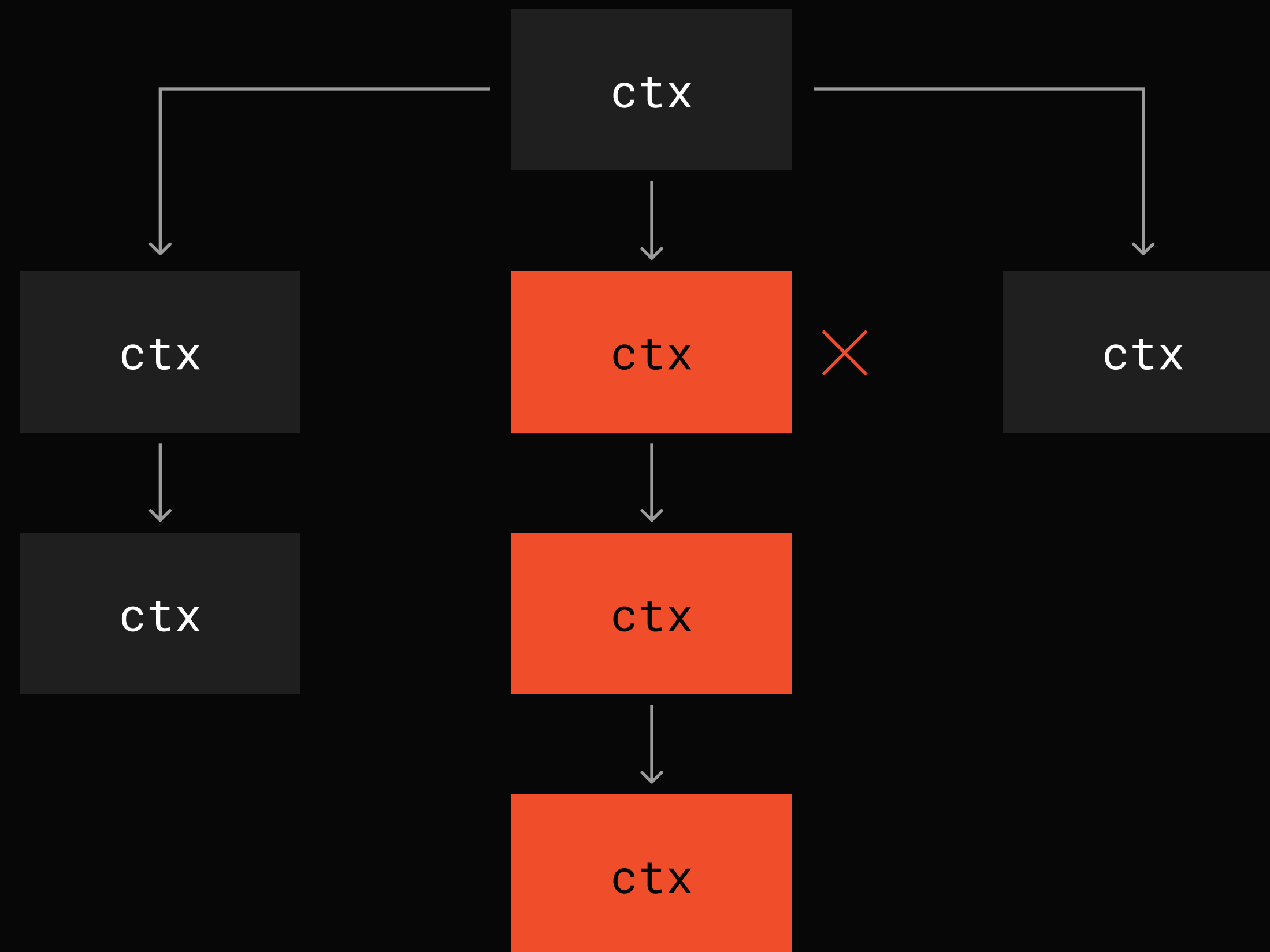
EXAMPLE

Context with timeout implementation

EXAMPLE

Context inheritance first

КОНТЕКСТ ИМЕЕТ ПРИРОДУ МАТРЕШКИ!



EXAMPLE

Context inheritance second

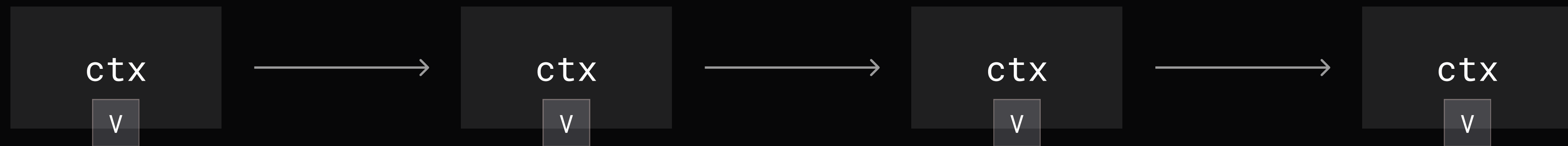
EXAMPLE

Context with value

EXAMPLE

Context with value inheritance

Use context Values only for request-scoped data that transits processes and APIs, not for passing optional parameters to functions



EXAMPLE

Context with value type

Две функции из разных пакетов в качестве ключа могут использовать одно и тоже значение, чтобы избежать коллизии – **рекомендуется использовать отдельный неэкспортируемый тип**

EXAMPLE

Context with cancel cause

EXAMPLE

Context with timeout cause

EXAMPLE

Context without cancel

EXAMPLE

Context with http client

EXAMPLE

Context with http server

EXAMPLE

Context handling

EXAMPLE

Graceful shutdown

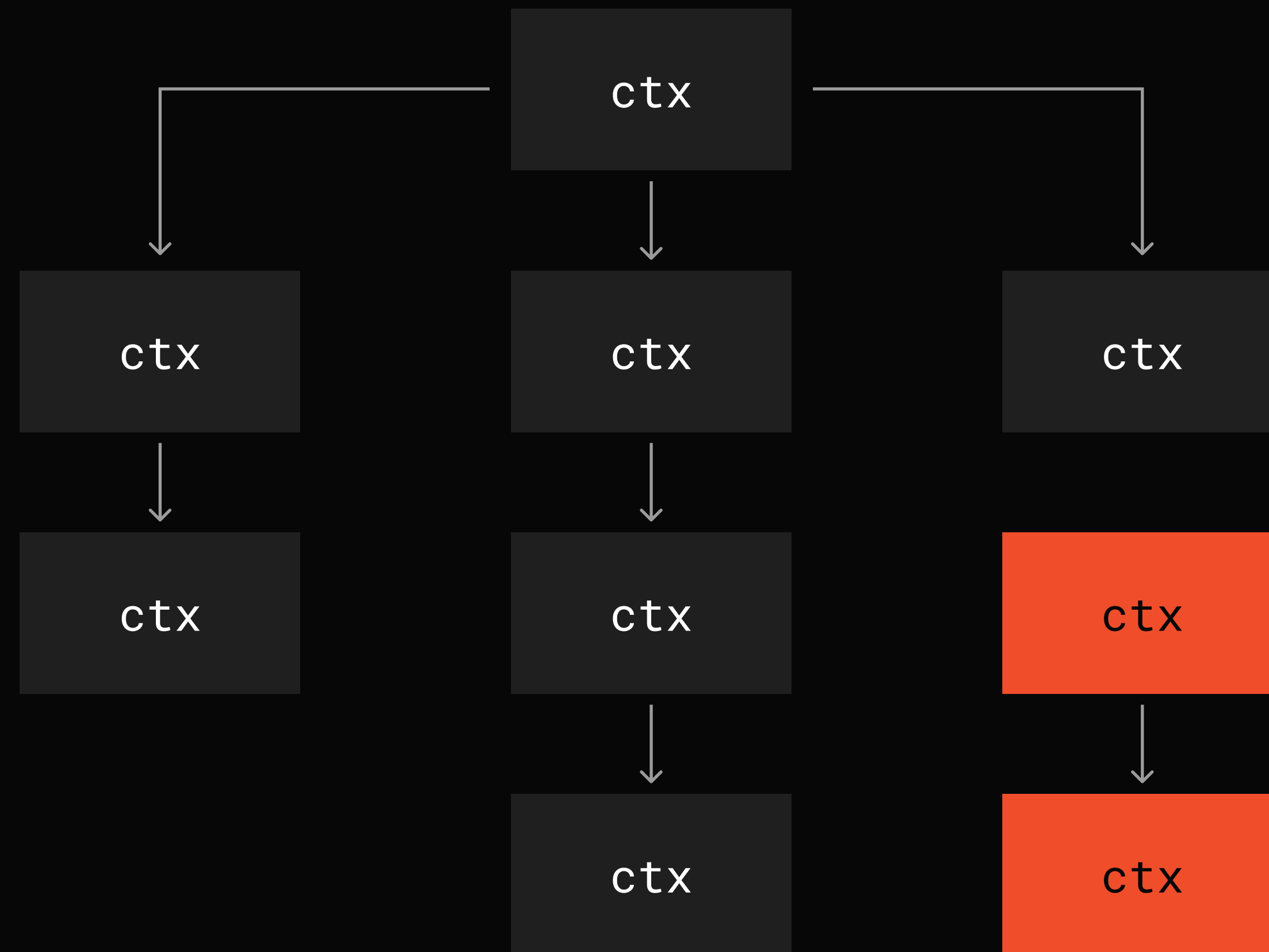
Передавайте контекст всегда первым аргументом —
это общепринятое соглашение

Передавайте только контекст, без функции отмены.

Контроль за завершением контекста должен
оставаться на вызывающей стороне, иначе логика
приложения может стать очень запутанной

Передавайте контекст только в функции и методы,
не храните в состоянии (внутри структуры) – контексты
спроектированы так, чтобы их использовали как
однообразовые и неизменяемые объекты

Используйте `context.WithValue` только в крайних случаях, в 99,9% случаев вы сможете передать данные через аргументы функции



context.Background должен использоваться только как самый
верхний родительский контекст, поскольку он является
заглушкой и не предоставляет средств контроля

Используйте `context.TODO`, если пока не уверены,
какой контекст нужно использовать

**Не используйте `nil` в качестве значения
контекста, вместо этого используйте `context.TODO`
или `context.Background`**

EXAMPLE

Nil context

EXAMPLE

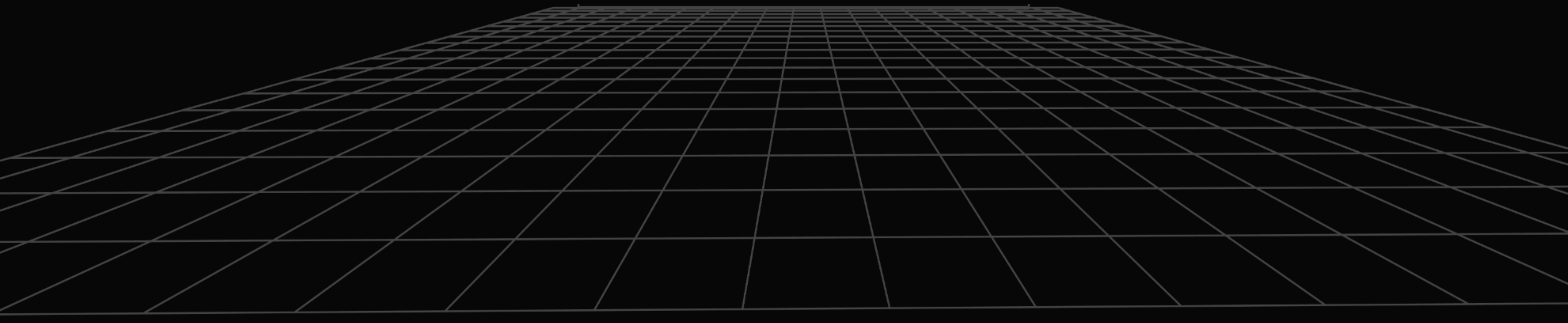
After done

EXAMPLE

Errgroup with context

FAQ

Контексты



ПОЖАЛУЙСТА, ЗАПОЛНИ ОПРОС О ЗАНЯТИИ

Ссылка в чате и в группе участников

