

# Loguru + Context

## + Log Persistence

*(Python ContextVar)*

# Problema

- Servicio basado en requests (web, Telegram bot...)
- Implementación de logs sobre varias funciones encadenadas



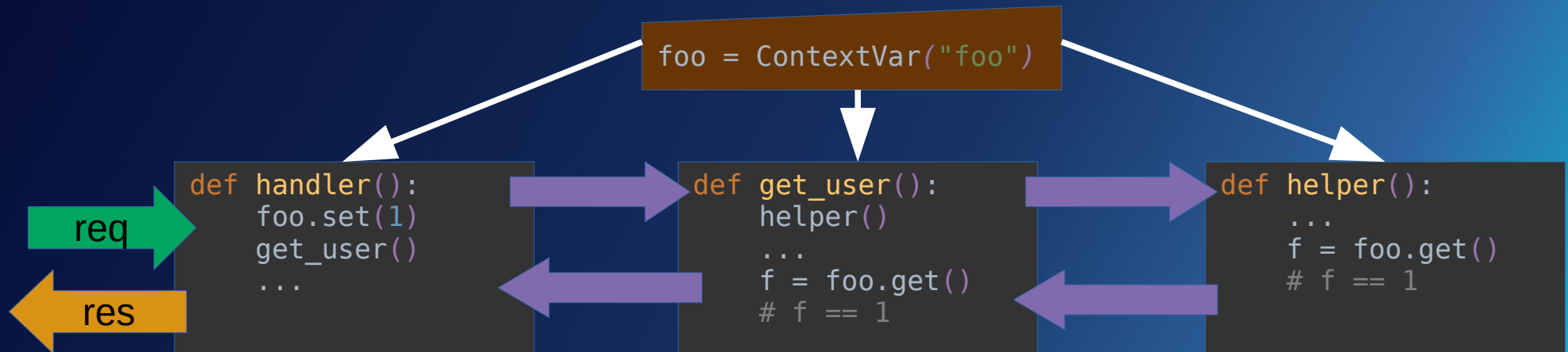
# Problema

- Servicio async
- Múltiples requests simultáneas
- Log records de varias requests mezclados

***¿Cómo saber el contexto de cada request  
en los logs?***

# Python ContextVars

- Se define ejecutando `set()` desde una función `A()`
- Todas las funciones llamadas desde `A()` pueden acceder al valor definido en `A()`



# Python ContextVars

- Nos permitirían añadir contexto a las líneas de logs

req

res

```
def request_handler():  
    req_id = str(uuid4())  
    context_req.set(req_id)  
  
    logger.debug(f"processing request {req_id}...")  
  
    get_user()  
    ...
```

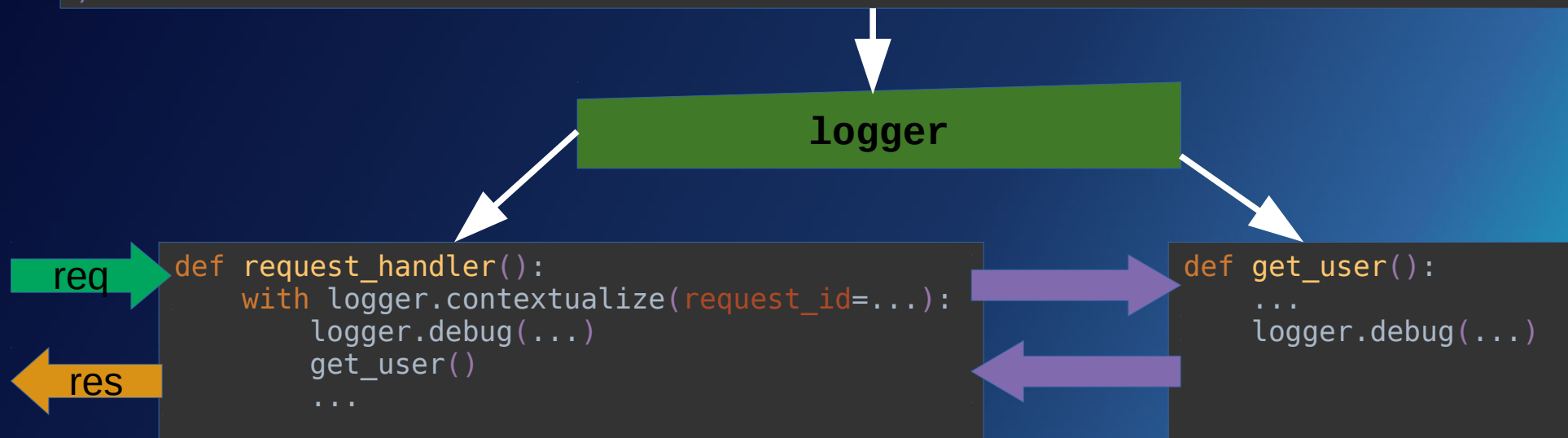
```
def get_user():  
    logger.debug(f"getting user (req={context_req.get()})")  
    ...
```

context\_req = ContextVar("req\_id")

# Logging context con Loguru

```
from loguru import logger

logger.remove()
logger.add(
    ...,
    level="DEBUG",
    format="{message} ({extra[request_id]})"
)
```

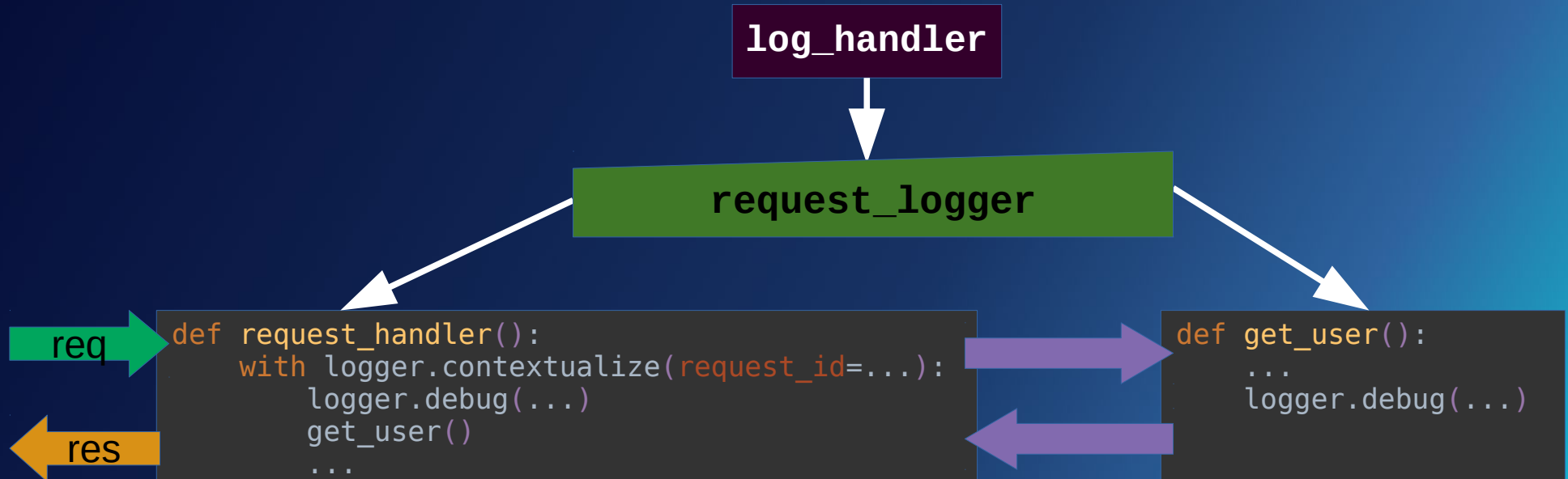


# Ejemplos de variables de contexto en logs

- UUID de la request
- Origen de la petición (IP, Telegram Chat ID...)
- Petición realizada (verb – endpoint, comando...)
- Variables de la petición (body/query params)

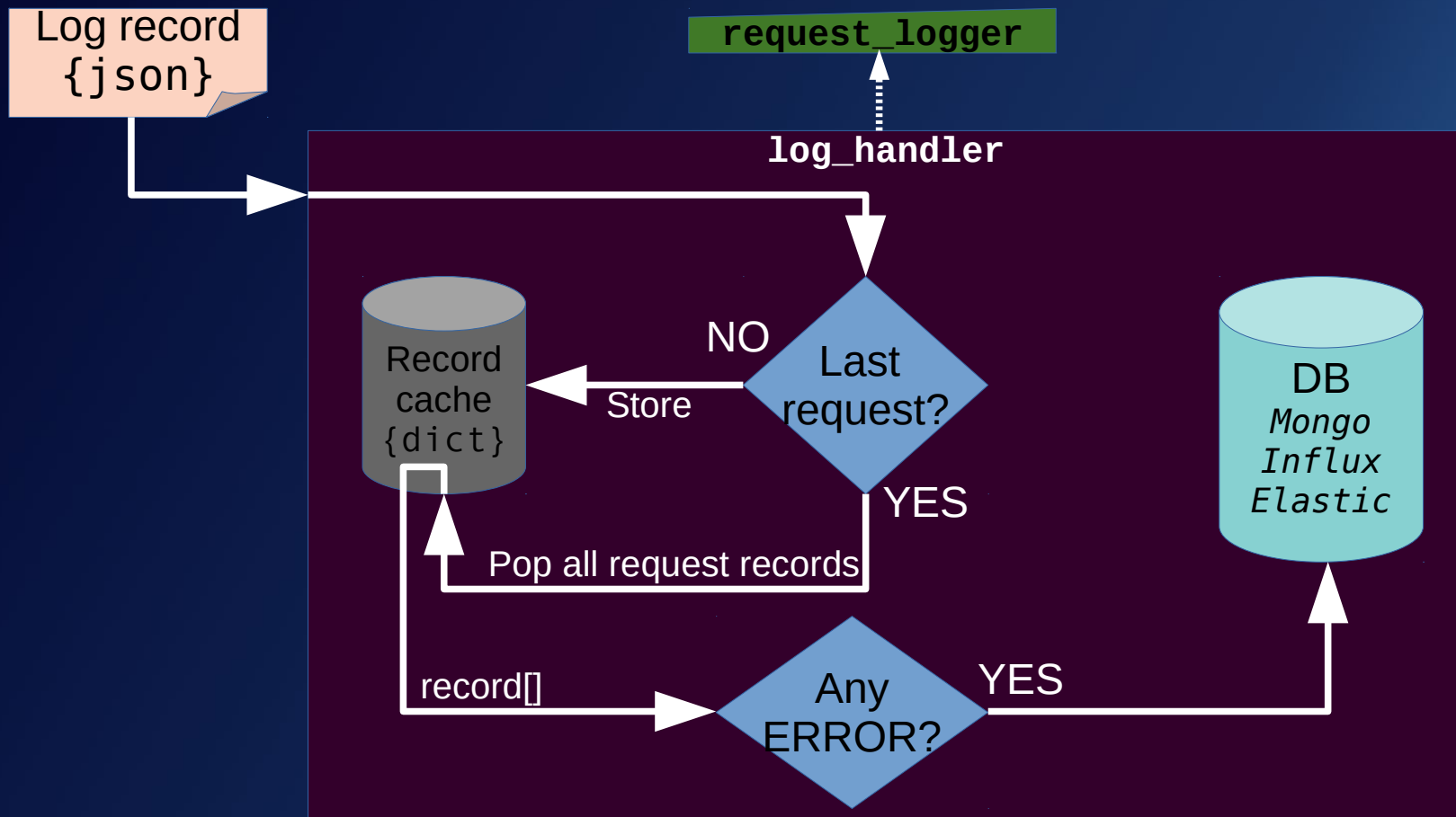
# Persisting (failed) requests

- Cuando una request falle, queremos persistir todos los log record originados durante esa request





# Persisting (failed) requests



(1) 374514c7-5ea1-49af-82a3-965253daf769	{ 2 fields }	Object
_id	374514c7-5ea1-49af-82a3-965253daf769	String
records	[ 3 elements ]	Array
[0]	{ 13 fields }	Object
[1]	{ 13 fields }	Object
elapsed	{ 2 fields }	Object
exception	null	Null
extra	{ 3 fields }	Object
file	{ 2 fields }	Object
function	post	String
level	{ 3 fields }	Object
line	58	Int32
message	Inserting new entity in DB: {'username': 'User that fails', 'user_id': '91a961d2-5991-11ea-ac9b-f00...	String
module	data_access	String
name	logging_requests_poc.data_access	String
process	{ 2 fields }	Object
thread	{ 2 fields }	Object
time	{ 2 fields }	Object
[2]	{ 13 fields }	Object
elapsed	{ 2 fields }	Object
exception	null	Null
extra	{ 4 fields }	Object
file	{ 2 fields }	Object
function	manage_endpoint_exceptions	String
level	{ 3 fields }	Object
line	33	Int32
message	Error while handling request: The username contains the word "fail" Traceback (most recent call l...	String
module	error_handler	String
name	logging_requests_poc.error_handler	String

# Ver más

- Repositorio + ejemplos:

<https://github.com/David-Lor/Loguru-Context-examples-lightning-talk>

- POC API:

<https://github.com/David-Lor/Logging-Requests-POC>

- Observability:

<https://en.wikipedia.org/wiki/Observability>

<https://www.scalyr.com/blog/observability-production-systems-why-how/>

<https://docs.honeycomb.io/learning-about-observability/intro-to-observability/>