BARCELONA **ZETTASCALE** LAB

Enabling the syscall_intercept library for RISC-V

Petar Andrić
Aaron Call
Ramon Nou
Guillem Senabre

System calls

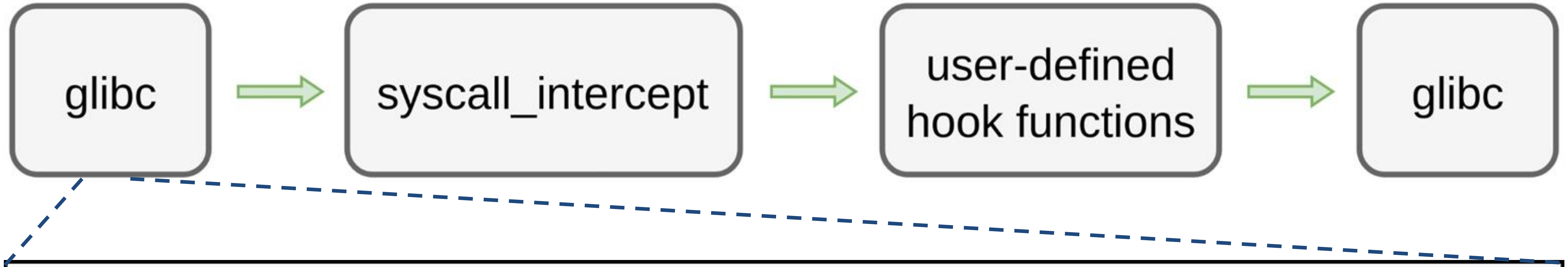
- The kernel manages privileged services such as devices, memory, file systems, etc.
- When an unprivileged program in user space wants to use a privileged service, it does so by calling the system (kernel), hence a system call.
- On RISC-V, this is accomplished via the ecall instruction, which stands for environment call. Based on the value stored in register a7, the kernel determines which specific system call to execute.
- The ecall instruction triggers a synchronous exception that raises the CPU privilege level, saves the current PC, and loads the kernel's trap handler address into PC.

System call interception

- It is a technique used to alter system-call behavior, enable sandboxing, perform dynamic analysis, inject errors for testing, and profile performance.
- We use it for GekkoFS, a highly scalable user-mode distributed file system for HPC clusters.
- There are many ways to perform system-call interception, such as eBPF, SUD, and ptrace. These rely on the kernel to manage the interception process, which makes them reliable but can negatively impact performance or provide limited control.
- The syscall_intercept library performs runtime binary rewriting, and the entire interception process is done in user mode, making it performant but not as reliable as kernel-managed interception methods.

The syscall_intercept library

```
$ LD_PRELOAD=./hooks.so bash -c 'echo $$'
```



<__getpid@@GLIBC_2.2.5>		
Address	Original	Patched
e3da0	mov eax,39	nop
e3da5	syscall	jmp <syscall_intercept>
e3da7	ret	ret

Obstacles on RISC-V

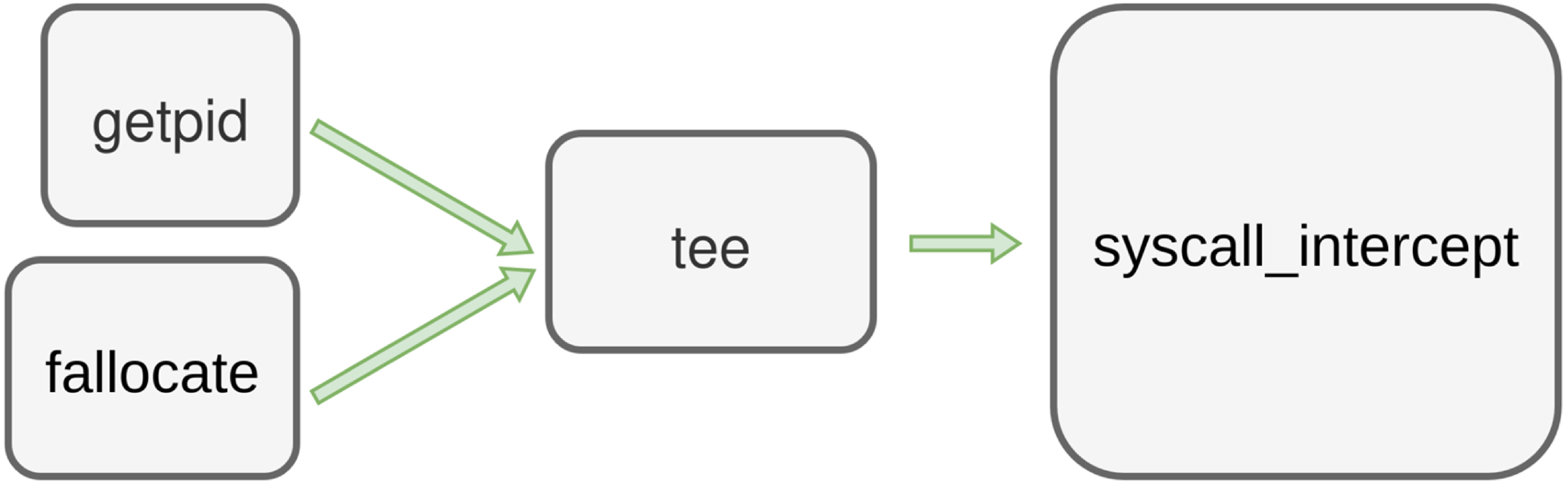
1. The relative jump instruction jal has a limited range.
2. RISC-V instructions are more compactly aligned than those on x86.
3. The Linux kernel on RISC-V preserves all general-purpose registers across a system call.

Architecture	Relative Jump	Reach	Size
x86	`jmp`	±2 GB	5 B
RISC-V	`jal`	±1 MB	4 B

Pseudo Patch Example on RISC-V	Size

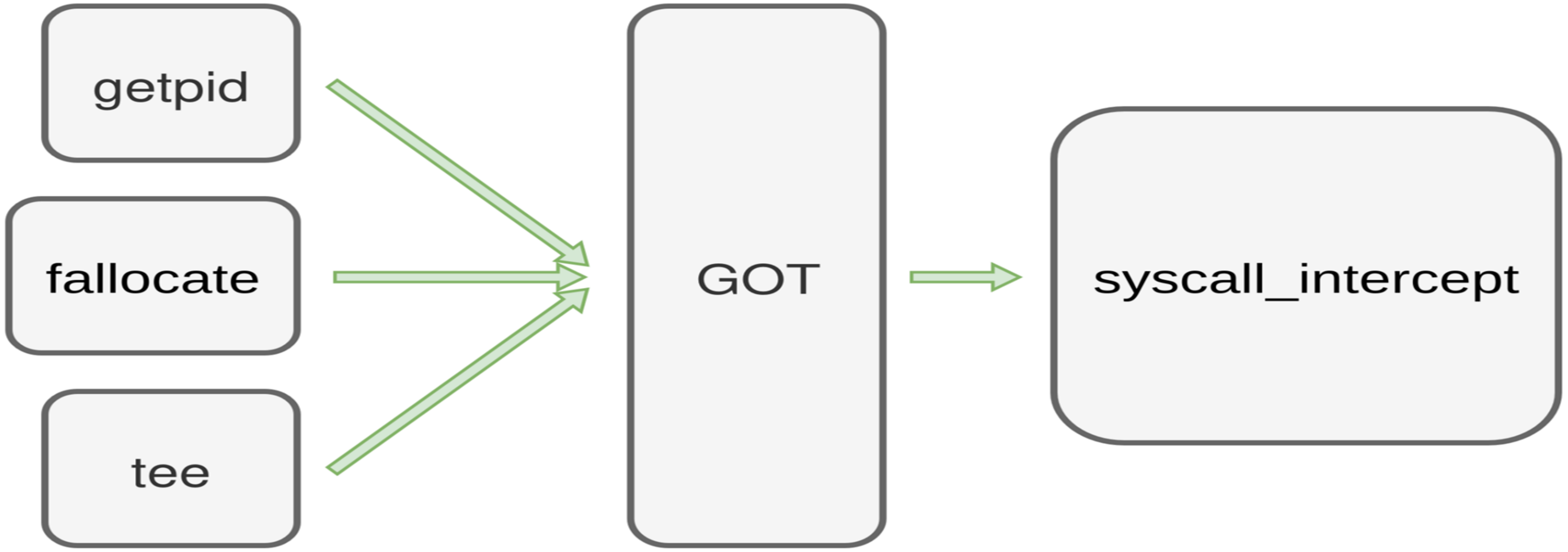
`Prologue`
`Indirect jump (auipc + jalr)`
`Epilogue`
	16 B

Gateway Solution



Gateway Patch: <tee@@GLIBC_2.27>				
Address	Original		Patched	
bb14e	mv	a2,s1	addi	sp,sp,-16
bb150	mv	a0,s0	sd	ra,0(sp)
bb152	li	a7,77	auipc	ra,<syscall_intercept>
bb156	ecall		jalr	ra,<off>(ra)
bb15a	lui	a4,0xfffff	ld	ra,0(sp)
bb15c	mv	s0,a0	addi	sp,sp,16

GPoline Solution



Complete Patch: <fallocate64@@GLIBC_2.27>				
Address	Original		Patched	
b3f84	li	a1,16	addi	sp,sp,-16
b3f86	mv	a2,sp	sd	ra,0(sp)
b3f88	ecall		jalr	gp
b3f8c	lui	a4,0xfffff	ld	ra,0(sp)
b3f8e	sext.w	a3,a0	addi	sp,sp,16

Improvements of the RISC-V Library

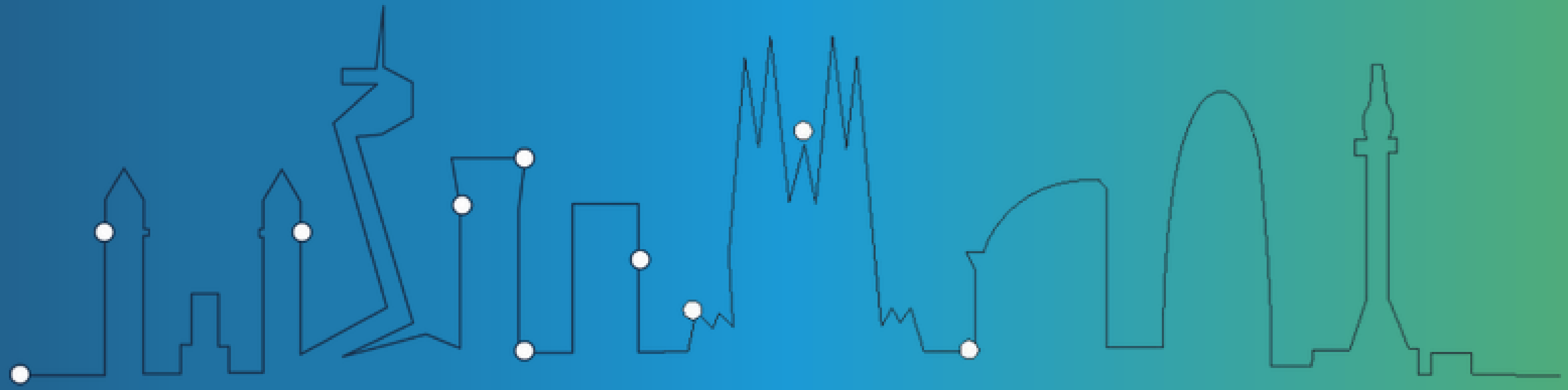
Improvements to the API compared to the x86 version:

1. No system call value is clobbered during interception (x86 clobbers rdx, while RISC-V preserves a1).
2. All threads (i.e., all clone variants) are intercepted, and their results are logged.
3. Post-clone hooks are triggered for every thread creation.
4. Two new environment variables allow filtering of patching system calls and shared libraries.

Performance and memory usage:
The x86 library dedicates a static block of instructions to each patch, while the RISC-V library uses the same entry point for every patch. This slightly increases processing time but significantly reduces memory usage and improves cache locality.

BZL

BARCELONA
ZETTASCALE LAB



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

BZL
BARCELONA **ZETTASCALE** LAB

MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia

Financiado por la Unión Europea
NextGenerationEU

Este proyecto con referencia REGAGE22e00058408992 está cofinanciado por el Ministerio para la Transformación Digital y de los Servicios Públicos, en el marco del Fondo de Resiliencia y Recuperación – y la Unión Europea – NextGenerationEU.

www.bzl.es