```jsx
import React, { useState, useEffect } from "react";
import Markdown from 'react-markdown';
import CodeBox from './CodeBox';

const DemoCard = props => {

        const { code, children } = props
        const enUs = 'en-US'
        const zhCn = 'zh-CN'
        const [markdown, setMarkdown] = useState('')

        useEffect(() => {
                let isMounted = true;
                fetch(code).then(res => res.text()).then(
                        md => {
                                if(isMounted) {
                                setMarkdown(md)
                                }
                        }
                );
                return () => { isMounted = false };
        }, [code]);

        return (
                <div className="code-box">
                        <section className="code-box-demo">
                                {children}
                        </section>
                        <section className="code-box-description">
                                {markdown && (
                                        <Markdown
                                                children={markdown}
                                                components={
                                                        {
                                                        h2: h => {
                                                                const isMdTitle = h.children[0].includes(enUs) && h.children[0].includes(zhCn);
                                                                const mdTitle = isMdTitle ? h.children[0] : '';
                                                                if (isMdTitle) {
                                                                        return <h4>{/en-US:(.+)/.exec(mdTitle)[1]}</h4>
                                                                }
                                                                return (
                                                                        <></>
                                                                )
                                                        },
                                                        hr : () => (
                                                                <></>
                                                        ),
                                                        p: p => {
                                                                return (
                                                                        typeof p.children[0] === 'string' && p.children[0].match(/[\u4e00-\u9faf]/) ? '' :<p className
                                                                )
                                                        },
                                                        pre: (pre) => {
                                                                const props = pre.children[0].props
                                                                const match = /language-(\w+)/.exec(props.className || '') || []
                                                                let language = ''

                                                                if (match.length > 0) {
                                                                        language = match[1]
                                                                }

                                                                return (
                                                                        <CodeBox language={language}>{props.children}</CodeBox>
                                                                )
                                                        }
                                                        }
                                                } />
                                )}
                        </section>
                </div>
        )
}

export default DemoCard
```