```rust
use std::str::Chars;

struct ConsIter<'a>{
    iter: Chars<'a>
}

struct Wrapper{
    inner: String
}

impl Wrapper{
    fn iter(&self)->ConsIter{
        ConsIter{iter: self.inner.chars()}
    }
}

impl<'a> Iterator for ConsIter<'a>{
    type Item = char;

    fn next(&mut self) -> Option<Self::Item> {
        while let Some(ch) = self.iter.next() {
            let lower = ch.to_ascii_lowercase();
            if ch.is_ascii() && !"aeiou".contains(lower) {
                return Some(ch); // Yield consonants and non-vowels
            }
        }
        None // End of iteration
    }
}
```