```
template <class E1, class E2> class wait_and{
  typedef typename E1:ret_type T1;
  typedef typename E2:ret_type T2;
  typedef std::tuple<T1, T2> ret_type;
 E1 & m1;
 E2 & m2;
  ...

  ret_type get(){
   return ret_type(m1.get(),m2.get());
  }

  template<class F>
  auto call_with_ret_vals(F && f)
   ->typename func_ret_type<f>::type{
      return f(m1.get(), m2.get());
  }

  template <class F>
  auto then(F && f)
   -> typename std::enable_if<
      is_callable_with_args<F, T1_t, T2_t>::value,
      func_ret_type<f>::type>::type {
         ...
  }
};
```